Queueing and Optimization Models for Hospital Patient Flow

Andersen, Anders Reenberg

Publication date:
2018

Document Version
Publisher's PDF, also known as Version of record

PhD Thesis

DTU Management Engineering
Department of Management Engineering

Queueing and Optimization Models for Hospital Patient Flow

Anders Reenberg Andersen

Kongens Lyngby 2018

DTU Management Engineering
Department of Management Engineering
Technical University of Denmark

Management Science
Produktionstorvet
Building 424
2800 Kongens Lyngby, Denmark
www.ms.man.dtu.dk

Assessment Committee

Stefan Røpke (Chairman)
Professor, Technical University of Denmark
Kongens Lyngby, Denmark

Richard J. Boucherie
Professor, University of Twente
Enschede, The Netherlands

Patrick De Causmaecker
Professor, KU Leuven
Kortrijk, Belgium

Main Supervisor

Thomas J. R. Stidsen
Associate Professor, Technical University of Denmark
Kongens Lyngby, Denmark

Co-supervisors

Bo F. Nielsen
Professor, Technical University of Denmark
Kongens Lyngby, Denmark

Line B. Reinhardt
Assistant Professor, Aalborg University
Copenhagen, Denmark

Summary

Various organizations claim that increasing attention should be put on an efficient use of healthcare resources. The internationally rising life expectancy and population size is accompanied by hospitals that are relying more on short admissions, and thus on limited bed capacity. The international World Health Report published by the World Health Organization shows that 20-40% of all healthcare resources are not being sufficiently utilized. Thus, tools that benefit an efficient healthcare system is greatly relevant to the present society.

The goal of this thesis is to expand methods in the field of modeling and optimizing hospital patient flow with a view to provide management and planners with a range of decision tools for improving the utilization of hospital resources. We elaborate on a number of relevant hospital optimization problems which relate to decision making on both the strategic, tactical and operational level. In addition, we focus on various types of patient flow, from inpatient to acute and surgical admissions, which has led to four different research studies.

Methodologically we mainly focus on evaluating the different instances of patient flow based on Markov chain modeling, and employing these models in heuristic search procedures to optimize the configuration of the related hospital resources. We employ this general approach in three studies. Additionally, the fourth study elaborates on a simulation-based Markov decision process. All four studies have been validated with patient data from Danish hospitals.

The thesis consists of seven chapters which have been divided into four different parts. The first part consists of two chapters, where Chapter 1 introduces the reader to the concept of hospital patient flow, and presents the motivation for modeling and optimizing the processes that are related hereto. Next, Chapters 2 prepares the reader for the methods that have been employed in our research with particular focus on Markov chain modeling and heuristic optimization.

Part II and III contain our contribution to the literature and comprise two chapters each. In Part II we focus exclusively on inpatient flow. Here, Chapter 3 presents a Markov chain model for evaluating the flow of inpatients, and a heuristic search procedure for deriving an improved distribution of the hospital's bed resources. By employing a heuristic statistical test we find that our approach adequately reflects the behavior of inpatient flow for a specific hospital case, and through additional tests that patient relocations can be reduced by 11.8% by re-distributing resources that are already available to the hospital.

Next, in Chapter 4 we extend the application of the Markov chain model by introducing patient preferences for room types into the optimization problem. That is, our goal is to maximize the number of patient preference-matches by changing the configuration of room types for the hospital wards. To achieve this we employ a randomized and interpolated search procedure, where solutions are sampled based on an interpolation between the currently known solutions in the search space. Numerical experiments show that this approach is able to derive near-optimal solutions usually within a 1% relative gap from

the optimum.

In Part III we focus on both acute and surgical patient flow. Chapter 5 presents a method for optimizing emergency department staffing by evaluating the patient flow as a Markov chain model. We employ this model in a search procedure that exploits integer linear programming to minimize the total amount of staff by simultaneously accounting for the patient waiting time. Simulation experiments indicate that our approach is fairly robust to our model assumptions, and that the solutions perform well in emergency departments with multiple triage-classes of patients.

Next, in Chapter 6 we present an approach for minimizing the long-term costs related to day-to-day scheduling of surgical patients. Here, we account for the inherent rolling horizon in the problem by employing a simulation-based Markov decision process. By using data from a hospital case, we validate the approach through various simulation experiments, which indicate that distinct improvements can be achieved by employing our approach rather than performing patient scheduling manually.

Finally, Part IV comprises a single chapter, namely Chapter 7, where we summarize the findings from each of our studies in a final conclusion to the thesis. In relation hereto, we provide the reader with our reflections and suggestions for future work.

Sammenfatning (Summary in Danish)

Flere organisationer hævder, at der skal være et større fokus på effektiv anvendelse af de ressourcer, som går til sundhed. Den verdensomspændende voksende gennemsnitslevealder og det stigende befolkningstal er ledsaget af hospitaler, som sætter deres lid til korte indlæggelser, og derved begrænset sengekapacitet. I den internationale World Health Report som er udgivet af World Health Organization, beskrives det, hvordan 20-40% af alle sundhedsressourcer ikke bliver tilstrækkeligt udnyttet. Derfor er alle værktøjer, som kan bidrage til et effektivt sundhedssystem, i høj grad relevante for nutidens samfund.

Målet med denne afhandling er at udbygge metoderne indenfor modellering og optimering af hospitalers patientflow med henblik på at tilvejebringe en række beslutningsværktøjer, som kan benyttes af både ledelse og planlæggere til at forbedre hospitalernes ressourceudnyttelse. Vi går i dybden med et antal relevante optimeringsproblemer, som relaterer sig til både det strategiske, taktiske og operationelle niveau. Herudover fokuserer vi på flere forskellige typer af patientflow fra langtidsindlæggelser til akutte og kirurgiske indlæggelser, hvilket har ført til fire forskellige forskningsstudier.

Metodisk fokuserer vi hovedsageligt på at evaluere de forskellige eksempler på patientflow med Markovkæder og dernæst at anvende disse modeller i heuristiske søgeprocedurer for at optimere konfigurationen af de relaterede hospitalsressourcer. Denne generelle tilgang bliver anvendt i tre studier. Hertil beskriver det fjerde studie en simulationsbaseret Markovbeslutningsproces. Alle fire studier er blevet valideret med patientdata fra danske hospitaler.

Afhandlingen består af syv kapitler, som er delt op i fire forskellige dele. Den første del består af to kapitler, hvor kapitel 1 introducerer læseren til konceptet omkring hospitalernes patientflow og uddyber motivationen for at modellere og optimere de processer, som er relateret hertil. Herefter forbereder kapitel 2 læseren på de metoder, som vi har anvendt i vores forskning med særlig fokus på Markovkæder og heuristisk optimering.

Del II og III indeholder vores bidrag til litteraturen og omfatter hver to kapitler. I del II fokuserer vi udelukkende på langtidsindlæggelser. Her præsenterer kapitel 3 en Markovkædemodel til at evaluere flowet for langtidsindlæggelser og en heuristisk søgeprocedure til at udlede en forbedret fordeling af hospitalers sengeressourcer. For et specifikt hospital viser vi ved hjælp af en heuristisk statistisk test, at vores tilgang reflekterer patienternes flow og gennem yderligere tests, at flytning af patienter kan reduceres med 11,8% ved at omfordele de ressourcer, som allerede er tilgængelige for hospitalet.

Herefter udvider kapitel 4 anvendelsen af Markovkædemodellen ved at indføre patienternes præferencer for rumtyper i optimeringsproblemet. Det vil sige, målet er at maksimere antallet af patient-præference-par ved at ændre konfigurationen af rumtyper for hospitalets sengeafsnit. For at opnå dette anvender vi en randomiseret og interpoleret søgeprocedure, hvor stikprøver af løsninger bliver udtaget baseret på en interpolation mellem de løsninger, som

allerede er kendt. Numeriske eksperimenter viser, at denne tilgang er i stand til at udlede løsninger, som er tæt på optimale. Det vil sige med en relativ forskel, som ofte er under 1% fra optimum.

I del III fokuserer vi på både akut og kirurgisk patientflow. kapitel 5 præsenterer en metode til at optimere bemandingen på en akutafdeling ved at evaluere patientflowet som en Markovkædemodel. Vi anvender denne model i en søgeprocedure, der benytter lineær heltalsprogrammering til at minimere den samlede bemanding ved at tage højde for patienternes ventetid på samme tid. Eksperimenter med simulering indikerer, at vores tilgang er temmelig robust i forhold til vores modelantagelser, og at løsningerne yder udmærket i akutafdelinger med flere triage-niveauer.

Herefter præsenterer kapitel 6 en tilgang til at minimere de langvarige omkostninger, som relaterer sig til den daglige skedulering af kirurgiske patienter. Her tager vi højde for problemets iboende rullende planlægningshorisont ved at anvende en simulationsbaseret Markovbeslutningsproces. Ved hjælp af data fra et hospital validerer vi tilgangen gennem forskellige eksperimenter med simulering, hvilke indikerer, at bemærkelsesværdige forbedringer kan opnås ved at indføre vores tilgang fremfor at udføre patientskedulering manuelt.

Endelig omfatter del IV et enkelt kapitel, nemlig kapitel 7, hvor vi opsummerer fundene fra hver af vores studier i afhandlingens endelige konklusion. Herudover kommer vi med vores reflektioner og forslag til fremtidig forskning.

Preface

This PhD thesis was carried out at the Department of Management Engineering at the Technical University of Denmark (DTU) and constitutes a collaboration between the university and the Danish hospitals. The research in this thesis has been co-funded by the governmental organization Region Sjælland, which is the public provider of healthcare on Lolland, Falster and the western part of Zealand, in Denmark. The organization manages seven public hospitals, and strives to continually improve their processes by introducing the use of mathematical modeling to both model and optimize their operations. For this reason, Region Sjælland has been collaborating with DTU on several educational projects, including both Master's thesis projects and full PhD educations, since 2014. This thesis constitutes the first PhD thesis in this collaboration.

In Denmark, hospital processes are already assessed with a view to improve both patient care and production rates. In Region Sjælland, one of the main contributors to these improvements is the department of Production, Research and Innovation (in Danish: Produktion, Forskning og Innovation) that supports hospitals in gaining insight into their processes by providing various analyses and decision support tools. A substantial amount of these analyses are based on patient data, which is logged and stored for every arriving patient. Besides finance, the department has provided much of the data for this thesis, which together with interviews and observations form the basis for the hospital cases that were investigated.

As a result, we have been able to scope our research on expanding the current literature on hospital planning, as well as solving problems that are encountered by hospitals on a daily basis.

The thesis contains a total of seven chapters with research that has lead to four journal articles. At this point, one article has been published, and one is accepted for publication. The main supervisor of the thesis is Associate Professor Thomas Jacob Riis Stidsen from the Department of Management Engineering at the Technical University of Denmark, and the co-supervisors are Professor Bo Friis Nielsen from the Department of Applied Mathematics and Computer Science at the Technical University of Denmark, and Assistant Professor Line Blander Reinhardt from the Department of Mechanical and Manufacturing Engineering at Aalborg University.

We hope that this thesis provides a basis for further research, and that it can be employed as support for management and planners in the hospital industry.

Kongens Lyngby, June 30th 2018

Anders Reenberg Andersen

Acknowledgments

This thesis would not have been without the support of many helpful people. First of all, the fundamental basis of this PhD project is the financial support by Region Sjælland, and director Mahad Huniche who came up with the idea of establishing a collaboration with the Technical University of Denmark.

Besides financial support, Region Sjælland provided the data and insight that the research in this thesis is largely based upon. This data would not have been employed if it were not for the talented people in the department of Production, Research and Innovation (in Danish: Produktion, Forskning og Innovation). Additionally, the staff at various hospital departments have been helpful in providing detailed insight into the hospital processes, which has been the basis of developing and validating the models in this thesis. In particular, I would like to thank the staff at Slagelse hospital, and especially the physicians and nurses at the emergency department for letting me observe their workflow.

In addition, I would like to thank my hosts at the KU Leuven Technology Campus in Ghent for providing me with an interesting and educational stay. I am thankful to have been under the supervision of Professor Greet Vanden Berghe, and to have worked with Postdoctoral Researcher Wim Vancroonenburg. They are both very skilled researchers as well as thorough authors.

Last, but certainly not least, I would like to thank my three supervisors for supporting me throughout my PhD education, and to have pushed me, but simultaneously given me sufficient space to choose the methods, the depth and the direction that I wanted to aim for. My main supervisor Associate Professor Thomas J. R. Stidsen has been supporting me with strategic decisions as well as mediating between me and Region Sjælland, which has ensured a steady flow in the project, and certainly limited my level of stress. Assistant Professor Line B. Reinhardt has been essential in all aspects of this project, among other things by making sure that the project was initialized to begin with, engaging in my research, and setting me up with various contacts. The same applies to Professor Bo F. Nielsen, who has been very engaged and detail-oriented with respect to my research, and a thorough co-author. Bo supported me in my Master's thesis, and continued his support throughout my entire PhD project, even though he was not added as an official supervisor until 2017. I am very grateful to have been under the supervision of all three supervisors!

Contents

Part I

Introduction

Chapter 1

Introduction

1.1 Motivation

Expenses related to healthcare constitute an ever increasing fraction of the Gross Domestic Product (GDP) for countries worldwide [55]. Statistics from the Organisation for Economic Co-operation and Development (OECD) show that the increasing investments are accompanied by a larger availability of physicians, whereas other essential healthcare resources are becoming increasingly more limited to the population. Furthermore, the average relative bed availability has been drastically decreasing since 1980 for a corresponding distinct increasing life expectancy and population size [55, 140]. For Denmark specifically, bed availability has been reduced by 50% during the last two decades due to the hospitals relying on short admissions. In fact, the average length of stay has dropped roughly 20% through the last two decades [55], and in response hereto hospitals must continuously seek to improve their processes in order to maintain a proper quality of care.

A conservative estimate from the *World Health Report*, published by the World Health Organization (WHO) [98], shows that 20-40% of all the resources that are dedicated to healthcare are being wasted. In order to reduce this waste, the report emphasizes that not only must sufficient funds be raised, but the efficiency and equity of the health system must be consolidated.

Returning to the utilization of resources in Danish healthcare, in 2014 Bloomberg published a list of the most efficient healthcare systems [26], where Denmark is ranked as number 34 out of 51 countries. In comparison, the neighboring country of Sweden is given a rank of 19, and Germany a rank of 23. In addition, a number of sources, of which some are healthcare professionals, state that the Danish healthcare system should be more in focus. In Pedersen & Petersen, 2014 [101] a variety of steps to improve hospital patient flow is described, and in Højgaard, 2017 [69] some of the essential problems related to both hospital staff and patient treatment is elaborated. Both of these sources are related to the Danish public healthcare sector. In addition hereto, a number of recent sources discuss how the Danish system is currently managed [70, 108, 86], thus all things considered indicating that development in the areas of management, mathematical modeling and optimization of hospital processes are greatly relevant to the present society [22].

Any hospital that is able to collect data that describes patient flow might be able to devise a mathematical (or computational) understanding of the internal processes, as well as how these can be modified to affect patient treatment. Naturally, this greatly depends on the quality of the observations, and the foundation unto which the collection-process is build [80]. For instance, if the data is logged by hospital staff that simultaneously have to prioritize the treatment of their patients, the credibility of the data might be affected.

In Denmark, collection of patient data have been mandatory through several years, and despite criticism of the collection process [71, 18], this thesis will show how patient data can be exploited to provide a valuable insight into modeling and optimization of patient flow.

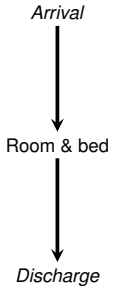
1.2 Hospital Patient Flow: A Brief Introduction

Patient flow is a diverse concept that features many different types of both private and governmental organizations. Some of these include, but are not confined to: General practitioners, rehabilitation in the municipalities, home care for the elderly, and blood banking. In this thesis we direct our attention to patient flow that relates to admissions or other visits at a hospital. By the term *flow*, we refer to the notion of modeling the utilization of resources that are associated with the patients during their stay at the hospital. Because resources are often utilized in a successive order they can also be interpreted as steps forming a path for the patient. Certain steps in these paths are repeated a substantial number of times yielding a *pattern* or *system* that yields the patient flow.

In this thesis, we base our interpretation of hospital patient flow on observations and interviews from hospitals in Denmark. More specifically, we have focused on hospitals that are managed in western Zealand by the governmental organization *Region Sjælland* [6]. The following will be based on Hall, 2006 [65] as well as our data from these hospitals.

Returning to the notion of patients "flowing through" the hospital. Notice that in the context of this thesis, patient flow refers only to the utilization of resources, and does not necessarily refer to a physical location shift. For instance, a patient might be physically located in the same room during which the patient have previously been examined by a physician, and is now waiting for a treatment. Suppose these two steps are sufficient to discharge this patient. The associated patient flow can then be depicted as presented in Figure 1.1a. At the same time, hosting the patient at the hospital requires a room and a bed. Thus, in the context of utilizing structural and physical resources (omitting the equipment needed to conduct the examination and treatment), we may consider the flow presented in Figure 1.1b.

(a) Flow related to procedures during the patient's length of stay.



(b) Flow related to hosting the patient.

The resources associated with the flow presented in Figure 1.1a are essentially independent of the flow presented in Figure 1.1b. Nonetheless, the physical resources of Figure 1.1b are often related to the organizational structure of the hospital, which is further related to the required treatment, finally leading to the procedures that a patient will be subject to, i.e. the flow in Figure 1.1a. However, there are exceptions to the relation between treatment and which resources that are employed to host the patient — for instance, patients that have been relocated due to insufficient bed capacity (cf. Chapter 3).

In order to control the progression of the different paths, *hospitals* use three aggregated patient classifications [65, p. 14]. Patients are grouped into these depending on their current state with a view to control the overall amount and types of resources that are employed. These patient classifications are often referred to as:

1. Inpatients
2. Outpatients
3. Emergency (Acute)

Inpatients Patients that stay at the hospital for more than one day and as a result require a bed are considered *inpatients*. This type can constitute a quite small fraction of the total number of patients that visit a hospital on a daily basis, but conversely they account for the vast majority of the hospital expenses.

Inpatients usually originate from the emergency department; otherwise they are transferred from a different institution (e.g. another hospital), or from the general community (e.g. a general practitioner). After arrival the inpatients will be admitted in a *ward*, where they will receive care and treatment during

their entire length of stay. Physically, inpatients will be located in a bed, and a room (cf. Figure 1.1b) that may be private or shared with other inpatients. From here, the patients will utilize both nursing and different *ancillary services* that requires both staff and equipment resources. These services might include the department of radiology, a pharmacy, or a lab.

Another important element of inpatient flow is surgery. Usually, surgical patients arrive to the hospital either as *elective* patients, meaning that they have been scheduled for an appointment, or as acute (through the emergency department), in which case they do not have an appointment, but require surgery immediately. A dedicated unit will ensure that the surgical procedure is performed after which a nursing ward or an intensive care unit (or both sequentially) will care for and monitor the patient until discharge [65, p. 14].

Outpatients Patients that have to visit the hospital, but do not have to stay overnight, are considered *outpatients*. This group may constitute the largest number of patients that make a visit to the hospital on a daily basis, and at the same time constitute an intermediate amount of the total hospital expenses.

Outpatients mainly originate from outside the hospital. That is, either they are transferred from a different hospital or institution, or they have been referred after visiting a general practitioner. As a result hereof, outpatients are often elective (scheduled) patients. Furthermore, as opposed to inpatients that are hospitalized in a bed, outpatients are scheduled to arrive at a unit referred to as an *outpatient clinic*. The hospital will comprise different types of clinics in accordance with the medical services they can offer. A typical path for an outpatient is a referral from a general practitioner, followed by the scheduling of an appointment at the hospital, where an examination is performed at an outpatient clinic. During the latter, the clinic will use resources from either a dedicated or centralized ancillary service similar to an inpatient path [65, p. 14].

Emergency Patients that require immediate care and, due to their condition, cannot wait for a scheduled appointment are considered *emergency* or *acute* patients. This group may constitute a medium number of daily admissions corresponding to about half of the outpatient arrivals. Additionally, acute admissions may account for a relatively small fraction of the total hospital expenses. Nonetheless, an efficient and well-functioning acute flow is essential to treating patients both in their current state, if they are transferred to an inpatient ward, or if they later return as an outpatient [99, 93, 72].

Acute patients originate from outside the hospital, either as walk-ins or by ambulance. On arrival, they are referred to a dedicated hospital unit, i.e. the *Emergency Department* (ED). According to our observations, the ED is sometimes notified of walk-ins in advance of the arrival, yielding a mix of both unknown (but to some extend predictable) and known incoming flow of patients. As a result of a mixed flow that contains patients of very different conditions, all new arrivals must undergo an initial examination known as a *triage*. During the triage, a member of the staff (e.g. a specialized triage nurse) evaluates

the condition of the patient, and assigns a code that specifies how the patient should be prioritized later in its path. In other words, whether the patient should be immediately attended by a physician or have to wait. To ensure an efficient flow through this process, the department will have pre-defined a number of parameters that govern exactly how the different triage codes should be used, and how they will segment the patients.

The remaining of the path for the acute patients varies from case to case. Some EDs rely on a single flow of patients, whereas others use parallel flows to ensure that patients with short treatment times can be attended without having to wait for the more severe cases. The literature sometimes refers to this as *fast track* flow [100, 43, 65].

In general, EDs are characterized by their short length of stay during which patients are stabilized and the remaining steps in their paths are determined. Certain acute patients may need to stay overnight, in which case they will usually be transferred to an inpatient ward. Acute patients are further inclined to return to the hospital at a later point as outpatients [65, p. 20].

All of the aforementioned patient types will be considered in different parts of this thesis. Inpatient flow will be considered in Chapter 3 and 4, emergency patients will be considered in Chapter 5, and in Chapter 6 we consider both in- and outpatients as they are scheduled for a surgical operation.

1.3 Overview of Related Research

In order to improve the processes that govern hospital patient flow, one needs to consider the methods that can be used to evaluate the system's performance, and in this regard, how the configuration of resources should be altered to enhance this performance. In this scheme, mathematical modeling and optimization plays a key role. In the remaining of this thesis, we will refer to the *modeling of patient flow* as the concept of creating mathematical insight into the behavior of the flow with a view to evaluate the system's performance. Conversely, the term *optimization* will be used in relation to deriving a solution to the configuration of the hospital resources, for instance by interacting with a model of the patient flow.

Naturally, patient flow modeling is an essential part of this thesis. A review by Bhattacharjee & Ray, 2014 [22] shows that three overall classes of approaches are used to model patient flow in the literature. Bhattacharjee & Ray refer to these as:

- Analytical
- Simulation
- Statistical (empirical)

The analytical class of approaches can be further divided into *queueing theoretic* models, and different applications of *Markov chain* models. Each of these methods attain different advantages depending on the nature of the problem at hand.

Queueing theory is a classical approach that has been employed to gain insight into hospital processes since the 1950s [17]. By queueing theory, we refer to an evaluation of patient flow performance by employing one or more analytical formulas. The specific measures that are usually considered are patient waiting times, overcrowding, and staff idle-time [22]. Expressing these measures as analytical formulas may seem beneficial, but there is a downside to this approach. By employing queueing theoretic models, one may have to consider a number of quite constraining assumptions. For instance, it may be necessary to assume that the system is in steady-state, and that the evolution of the process is Markovian [22]. We will elaborate more on these assumptions in Chapter 2.

A few specific cases of queueing theory are Gorunescu et al., 2002 [59, 60] and Li et al., 2009 [83] that employ two phase-type queueing models, $M/PH/c/N$ and $M/PH/c$, to model mixed patient flows. In addition hereto, Green, 2002 [62] employ an $M/M/c$ model to evaluate the availability of beds for a number of different hospital units. Lastly, open queueing networks are employed by Cochran & Roche, 2009 [42] and Mayhew & Smith, 2008 [88] to investigate how patient throughput can be increased.

Compared to the queueing theoretic models, the literature is scarce when it comes to research in Markov chains for modeling patient flow [22]. Even though Markov chain modeling is closely related to queueing theory, the approach has been widely used to replace statistical models, for instance by modeling the length of stay for patients. Specifically, discrete-time Markov chains have been employed by Bartolomeo et al., 2008 [19] to model the readmission probability of patients, and by Broyles et al., 2010 [29] to predict the number of inpatients in a hospital. Furthermore, a continuous-time Markov chain is used by Shaw & Marshall, 2007 [116] to model the length of stay for heart-failure patients, whereas Wang et al., 2014 [135] evaluate the care delivery process of patients based on a closed network.

Computer simulations that are based on discrete event simulation, agent based simulation, and system dynamics have been extensively employed to model different types of healthcare systems, including patient flow [22]. Particular attention has been given to the unscheduled flows (e.g. acute patients), and reviews conducted by Lim et al., 2012 [85], and Borgman, 2017 [27] show that unscheduled patient flows are modeled by employing simulation in a substantially greater number of cases than the analytical methods. Bhattacharjee & Ray notice that the reason why simulation is a suitable choice, is due to the high complexity and time-dependent behavior of these specific flow types [22].

Specific applications of simulation include among others Khadem et al., 2008 [76] where a new layout for an emergency department is assessed by

employing a discrete event simulation, and Wang, 2009 [136] where different triage and radiology procedure settings are evaluated by using an agent based simulation.

The statistical, or empirical, models are only employed as the sole approach in few studies. Bhattacharjee & Ray characterize this group of models as the methods that are entirely based on observations and system experimentations with a view to analyze the dependencies in the patient flow [22]. They further state that this modeling approach is currently in its nascent stage. Examples include Adeyemi & Chaussalet, 2008 [9] and Adeyemi et al., 2011 [8] where random effects models are used to identify patient pathways.

In this thesis, we generally focus on employing Markov chains to model patient flow, and as we have shown in the above, as well as in our subsequent literature reviews (cf. Chapter 3-5), the use of Markov chains to model patient flow is an uncommon approach. Furthermore, few studies seem to exist where algorithmic optimization is conducted by employing the analytical methods, mentioned above, to evaluate the effects on the patients flow [85]. By algorithmic optimization we refer to methods such as heuristic and matheuristic search procedures. In fact, the studies that conduct optimization based on this type of algorithms tend to rely on simulation [119, 30].

1.3.1 Contribution

Our contribution to the current literature on patient flow modeling and optimization will be carefully clarified throughout Chapter 3-6. By summarizing, we provide:

- A Markov chain for modeling inpatient flow that accounts for patient relocation. We demonstrate how to cope with the "curse of dimensionality" for this model by truncating the state space, and how to validate the model using patient data.
- A heuristic and a matheuristic search procedure for optimizing bed and room resources for inpatient flow, respectively. The first is based on a hill climber heuristic, and the second on an interpolation of the known solution samples in the search space. Later, we shall refer to the latter as *randomized and interpolated search*. The solutions in both search procedures are evaluated by employing a Markov chain model.
- A Markov chain for modeling time-dependent acute patient flow. We validate the model by comparing to a number of simulations that feature different service time distributions as well as patient classes.
- A matheuristic search procedure for optimizing the staffing of an emergency department. In this search procedure, we recursively allocate staff to the department by using an integer linear programming model.

The search procedure ensures not to violate any constraints on patient waiting time by employing a Markov chain model of the system.

- A simulation-based Markov decision process for optimizing the scheduling of surgical patients with a rolling horizon. We demonstrate the performance of the model under different conditions, and compare our results to a number of other scheduling policies.

1.4 Thesis Outline

The content of this thesis is divided into four parts. The current Part I, introduces the reader to the area of patient flow modeling and optimization, and consists of two chapters. Furthermore, Part II and III contain our contribution to the literature and comprise four articles. Part II is dedicated to long-term admissions, or inpatient flow, and Part III is dedicated to more short-term admission, i.e. acute and surgical flow. Each of these parts contains two chapters, where each chapter represent a specific hospital problem. Lastly, Part IV contains a single chapter with a final conclusion to the current thesis.

In the following, we present a brief description of the content of each remaining chapter.

Chapter 2 introduces the methods that have been used to both model and optimize patient flow throughout this thesis. More specifically, the chapter will be divided into two sections: The first section contains a description of the methods related to modeling the behavior of patient flow, i.e. a brief introduction to the fundamentals of queueing theory followed by an introduction to Markov chain modeling. The second section contains an introduction to heuristic optimization, where a number of basic concepts are presented, followed by a few specific examples of the matheuristic search procedures that have been employed in this thesis. The purpose of this chapter is to prepare readers with various backgrounds for the subsequent chapters.

Chapter 3 presents a method for modeling inpatient flow and optimizing the associated distribution of bed resources. More specifically, in this chapter we consider a hospital problem featuring a set of wards and corresponding set of patient types that continuously require admission at the hospital. On arrival, each type is dedicated to a specific ward in the system, unless the bed capacity of the ward has been depleted. If the latter is the case, then patients will either be lost from the system, or relocated to an alternative ward in the set of wards. We employ a continuous-time Markov chain to model this behavior, and a hill climber heuristic to optimize the distribution of resources for so to minimize the number of patients that are relocated (or lost) on arrival.

This chapter has been published in the European Journal of Operational Research with the title *Optimization of hospital ward resources with patient*

relocation using Markov chain modeling. The authors of the article are Anders Reenberg Andersen[1], Bo Friis Nielsen[2] and Line Blander Reinhardt[3] [13].

Chapter 4 extends the application of the Markov chain model from Chapter 3 by introducing room types to the optimization problem. That is, we consider a setting where patients are not only dedicated to a specific ward, but have preferences regarding the type of room that they are admitted to. We do not introduce any changes to the Markov chain, but exploit the occupancy distributions resulting from the model to define an objective function that accounts for the expected amount of patient preference-matches from the room configuration. We then maximize this function by employing a search procedure that recursively conducts random samples from the search space based on an interpolation between the currently known solutions.

This chapter has been submitted to Artificial Intelligence in Medicine with the title *Strategic room type allocation for nursing wards through Markov chain modeling*. The authors of the article are Anders Reenberg Andersen[1], Wim Vancroonenburg[4] and Greet Vanden Berghe[4].

Chapter 5 presents a method for optimizing the allocation of staff resources to an emergency department. More specifically, our aim is to derive the minimum amount of staff that is required to operate the department by simultaneously accounting for constraints on the patient waiting time. To achieve this, we model the occupancy of patients in the department as an open queueing network by employing a continuous-time Markov chain. Here, time-dependent behavior is evaluated by using the uniformization method. From the resulting state probability distribution, we evaluate the waiting time in each node of the network, which is then used as a constraint in a matheuristic search procedure. Furthermore, we validate our approach by comparing to several simulations of the associated system.

This chapter has been accepted for publication in the European Journal of Operational Research with the title *Staff optimization for time-dependent acute patient flow*. The authors of the article are Anders Reenberg Andersen[1], Bo Friis Nielsen[2], Line Blander Reinhardt[3] and Thomas Jacob Riis Stidsen[1].

Chapter 6 differs from the aforementioned Chapter 3-5 in the sense that we do not consider a queueing system, but instead the flow of surgical appointments. That is, in this chapter we investigate a method for scheduling appointments to specific dates and operating rooms, and simultaneously account for future surgical requests by modeling the problem with a rolling planning horizon. Specifically, our aim is to minimize the total long-term expected costs of

[1]Department of Management Engineering, Technical University of Denmark, Kongens Lyngby, Denmark

[2]Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kongens Lyngby, Denmark

[3]Department of Mechanical and Manufacturing Engineering, Aalborg University, Copenhagen, Denmark

[4]Department of Computer Science, KU Leuven, Ghent, Belgium

scheduling patients for a surgical operation, ensuring that patients can receive an appointment immediately, and that the procedure is conducted within a limited number of days. To achieve this, we employ a simulation-based Markov decision process in an online scheme which we refer to as rollout. Furthermore, since the action space is often intractable, we use a heuristic search procedure to construct an action, resulting in an allocation of the patients to the schedule. We compare this approach to a number of simple and advanced scheduling policies.

This chapter has been submitted to Annals of Operations Research with the title *Simulation-based rolling horizon scheduling for operating theatres*. The authors of the article are Anders Reenberg Andersen[1], Thomas Jacob Riis Stidsen[1] and Line Blander Reinhardt[3].

Chapter 7 summarizes the findings from each of the previous chapters in a final conclusion to the thesis. In addition, we provide the reader with our reflections and suggestions for future research in the area of patient flow modeling and optimization.

Chapter 2

Basics of Queues and Heuristic Optimization

2.1 Queues

To properly understand the behavior of hospital patient flow, we must distinguish between the situations in which patients are "in process" (e.g. examined, treated, recovering, etc.) and the situations in which they are simply waiting for some resource to become available. Often the paths that patients follow through a hospital can be viewed as a system, where the patients are either in a state of *service* or *waiting*. Patients might wait for a nurse upon arrival at the emergency department, or upon admission they might wait for a laboratory technician to collect a blood sample. If surgery is required, the patients will have to wait for an open slot in the calendar. In other words, patient flow is in many situations a system of queues that work as a network where nodes affect each other as patients switch between them. Performance measures such as length-of-stay and ward occupancy are a result of the underlying queueing network, and a function of the amount of available staff, the behavior of the staff, and the condition and rate at which patients arrive to the hospital. For this reason, understanding and modeling queues is an essential tool in this thesis.

In the following, we will present some of the fundamental elements of queueing theory, along with some of the elementary, but sometimes quite useful, queueing models. Next, in Section 2.1.1 and 2.1.2 we will present some of the more advanced methods that will be used to both model and validate queueing networks throughout this thesis.

Consider a stream of entities arriving at a node, within which they are processed, and then removed from the system (cf. Figure 2.1). The node can only take a certain number of entities and process them at the same time. For this reason, any entity that arrives when the capacity of the node is depleted has to wait, resulting in a *queue* emerging in front of the node. In some cases, the queue itself has a limited capacity causing the entities to be rejected. The reader might notice how the description of this system resembles a supermarket or a call-center, and for this reason standard theory [63, 123] often refers to the node as a service facility, the capacity of the node as servers, and the arriving entities as customers. In fact, queueing theory has been ap-

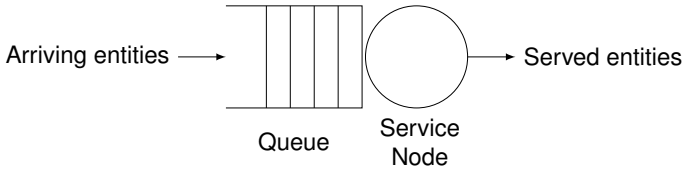


Figure 2.1: The basic structure of a queue.

plied extensively in these fields. For instance, the Danish mathematician and engineer Agner Krarup Erlang (1878-1929) is recognized for his pioneering research while working at the Telephone Company in Copenhagen [3]. Notice that the arriving entities might as well be patients, and the servers could be physicians.

The characteristics of a queueing system is to the most part determined by the following components:

- The behavior of the arriving patients.
- The behavior of the servers.
- The capacity of the system including the number of parallel servers.
- The queueing discipline.

The latter could be according to priority, such as in an emergency department where patients are triaged, and then examined according to their code. In addition, some hospital queueing systems treat patients (and other tasks) according to their time of arrival, which is known as First-In-First-Out (FIFO).

Arrival and Server Behavior

In order to model the arriving patients, we must consider the process that generates them. In this thesis, arrivals are always stochastic and generated according to a Poisson process. Thus, by letting $k \in \mathbb{N}_0$ define the number of arrivals in a time-interval of size $t \in \mathbb{R}_{\geq 0}$, we have that

$$Prob\{k \text{ within } t\} = p_k(t) = e^{-\lambda t}\frac{(\lambda t)^k}{k!} \tag{2.1}$$

where $\lambda \in \mathbb{R}_{>0}$ is the constant rate at which patients are arriving to the system. We now present some basic properties [123, p. 389] of (2.1).

Let $N(t) \in \mathbb{N}_0$ define the number of patients that arrive within the time-interval t; then for a Poisson process we have that $N(0) = 0$, patients arrive with independent increments, and the number of patients k depends only on the size of t and not on the past history of the system. Furthermore, for an increment of size $\Delta \in \mathbb{R}_{>0}$, we have that

- $Prob\{k = 0 \text{ within } (t, t + \Delta]\} = 1 - \lambda\Delta + o(\Delta)$

- $Prob\{k = 1 \text{ within } (t, t + \Delta]\} = \lambda\Delta + o(\Delta)$
- $Prob\{k > 1 \text{ within } (t, t + \Delta]\} = o(\Delta)$

where $o(\Delta)$ is a quantity that becomes negligible as $\Delta \to 0$ such that

$$\lim_{\Delta \to 0} \frac{o(\Delta)}{\Delta} = 0 \tag{2.2}$$

In the following we will demonstrate that a Poisson process results in an exponentially distributed time between each successive arrival. Consider $p_0(t + \Delta)$, namely the probability that no arrivals have occurred in the interval $(0, t + \Delta]$. Then, $p_0(t+\Delta) = p_0(t) \cdot Prob\{k = 0 \text{ within } (t, t+\Delta]\}$ which from the above yields

$$p_0(t + \Delta) = p_0(t) \cdot (1 - \lambda\Delta + o(\Delta))$$

$\Leftrightarrow$

$$p_0(t + \Delta) - p_0(t) = -p_0(t)\lambda\Delta + p_0(t)o(\Delta)$$

$\Leftrightarrow$

$$\frac{p_0(t + \Delta) - p_0(t)}{\Delta} = -p_0(t)\lambda + p_0(t)\frac{o(\Delta)}{\Delta}$$

which in the limit as $\Delta \to 0$ leads to

$$\frac{dp_0(t)}{dt} = -\lambda p_0(t) \Leftrightarrow \frac{dp_0(t)}{dt} + \lambda p_0(t) = 0 \tag{2.3}$$

according to (2.2). For a differential equation of the form $\frac{dy(x)}{dx} + \Phi(x)y(x) = \Psi(x)$, the solution is [63, p. 25],

$$y(x) = Ce^{-\int \Phi(x)dx} + e^{-\int \Phi(x)dx} \int e^{\int \Phi(x)dx} \Psi(x)dx$$

which by substituting with the elements of (2.3) leads to

$$p_0(t) = Ce^{-\int \lambda dt} + e^{-\int \lambda dt} \int e^{\int \lambda dt} \cdot 0 \cdot dt = Ce^{-\lambda t}$$

Applying the boundary condition $p_0(0) = 1$ we get $C = 1$, and thus,

$$p_0(t) = e^{-\lambda t} \tag{2.4}$$

Now, let the random variable $X \in \mathbb{R}_{>0}$ define the inter-arrival time of the patients with Cumulative Distribution Function (CDF) $F(t) = Prob\{X \leq t\} = 1 - Prob\{X > t\}$, corresponding to

$$F(t) = 1 - p_0(t)$$

Thus from (2.4), we finally get

$$F(t) = 1 - e^{-\lambda t}$$

which is the CDF for the exponential distribution. This type of distribution is *memoryless* [63, p. 29], also known as the *Markovian property* (from Andrei Andreevich Markov (1856-1922)), meaning that the time until the next arrival of a patient is independent of the time that has passed since the latest arrival.

In *Kendall's notation* [74], inter-arrival times that are exponentially distributed are defined by an M (referring to the Markovian property of the distribution). Thus, for a multi-server queueing system that has both exponential inter-arrival and inter-service times, we write $M/M/c$, which in short yields the distribution between first the arrivals, then the finished services, and last the number of parallel servers, $c \in \mathbb{N}_{>0}$. Naturally, real-life queueing systems can have many different types of distributions, such as deterministic (D), phase-type (PH), Erlang of type k (E_k), and general independent (GI or G).

Probability Distributions

A queue of the type $M/M/c$ can be viewed as a *birth-death* process. That is, a continuous-time Markov chain where the system can only change to the state $s \in \{0, 1, 2, \ldots, \infty\}$ through state $s-1$ or $s+1$. We elaborate more on Markov chains in Section 2.1.1. For the time being consider the process depicted in Figure 2.2, showing the transitions between states for the birth-death process associated with the $M/M/c$ queue, where each node represents the total number of patients that are either in queue or service. In other words, s denotes the number of patients in the system. This number increases with a rate corresponding to the arrival rate λ, and decreases with the rate $\mu_s = s\mu$, where $\mu \in \mathbb{R}_{>0}$ is the rate at which patients are treated at each server (i.e. the reciprocal of the mean inter-service time). However, this only applies as long as $s \leq c$; otherwise the rate is bounded at $\mu_s = c\mu$.

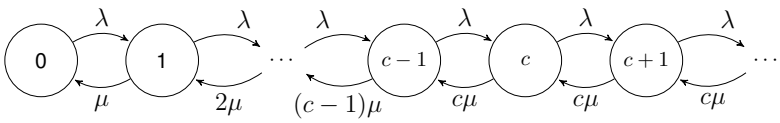


Figure 2.2: The state transitions associated with the $M/M/c$ queue.

In other words, for the rate at which the number of patients are discharged in the system, we have

$$\mu_s = \left\{ \begin{array}{ll} s\mu, & 1 \leq s \leq c \\ c\mu, & s \geq c \end{array} \right\}$$

This system is only stable if $\rho = \lambda/(c\mu) < 1$, at which point the probability of k patients occurring in the system is [123, p. 420],

$$p_k = p_0 \prod_{s=1}^{k} \frac{\lambda}{\mu_s}$$

Thus by employing our definition of rate μ_s, this leads to

$$p_k = p_0 \prod_{s=1}^{k} \frac{\lambda}{s\mu} = p_0 \left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!} \quad \text{if } 1 \leq k \leq c$$

and

$$p_k = p_0 \prod_{s=1}^{c} \frac{\lambda}{s\mu} \prod_{s=c+1}^{k} \frac{\lambda}{c\mu} = p_0 \left(\frac{\lambda}{\mu}\right)^k \frac{1}{c!} \left(\frac{1}{c}\right)^{k-c} \quad \text{if } k \geq c$$

We can simplify the above by employing the expression $\rho = \lambda/(c\mu)$, substituting λ/μ with $c\rho$. This leads to the final steady-state probabilities for the $M/M/c$ queue,

$$p_k = p_0 \frac{(c\rho)^k}{k!} \quad \text{if } 1 \leq k \leq c \tag{2.5}$$

and

$$p_k = p_0 \frac{(c\rho)^k}{c^{k-c}c!} = p_0 \frac{\rho^k c^c}{c!} \quad \text{if } k \geq c \tag{2.6}$$

In order to apply the above, we still need to derive p_0. Therefore, consider that we in general have that

$$\sum_{k=0}^{\infty} p_k = p_0 + \sum_{k=1}^{\infty} p_k = 1$$

Thus from expression (2.5) and (2.6), we can derive

$$p_0 \left(1 + \sum_{k=1}^{c-1} \frac{(c\rho)^k}{k!} + \sum_{k=c}^{\infty} \frac{\rho^k c^c}{c!}\right) = 1$$

$\Leftrightarrow$

$$p_0 = \left(1 + \sum_{k=1}^{c-1} \frac{(c\rho)^k}{k!} + \sum_{k=c}^{\infty} \frac{\rho^k c^c}{c!}\right)^{-1}$$

Furthermore, we can rewrite the infinite series

$$\sum_{k=c}^{\infty} \frac{\rho^k c^c}{c!} = \frac{1}{c!} \sum_{k=c}^{\infty} \rho^k c^c = \frac{(c\rho)^c}{c!} \sum_{k=c}^{\infty} \rho^{k-c}$$

which now yields the form $\sum_{i=0}^{\infty} ax^i = a/(1-x)$; hence

$$\frac{(c\rho)^c}{c!}\sum_{k=c}^{\infty}\rho^{k-c} = \frac{(c\rho)^c}{c!}\frac{1}{1-\rho}$$

and therefore

$$p_0 = \left(1 + \sum_{k=1}^{c-1}\frac{(c\rho)^k}{k!} + \frac{(c\rho)^c}{c!}\frac{1}{1-\rho}\right)^{-1} \tag{2.7}$$

Together, equation (2.5)-(2.7) yield the entire steady-state probability distribution of the $M/M/c$ queue. The upside of this simple queueing model is that it accounts for many real-life characteristics, such as multiple servers, Poisson generated arrivals, and inter-service time variability. However, it might occur to the reader that many queueing systems do not have exponential inter-service times (rather gamma or log-normal), in which case the $M/M/c$ might be inadequate to describe the behavior of the queueing system.

Consider a system that has a capacity-limit such that patients are lost from the system if all servers are occupied on arrival. That is, a queue of the type $M/M/c/c$, where the last c indicates that the capacity of the system is equal to the number of servers. In practice, this type of system occurs whenever patients do not wait for a resource (e.g. a nurse or a physician), but instead are relocated to a location where capacity is still available. Thus, the total number of patients in the system is defined as $s \in \{0, 1, \ldots, c-1, c\}$, so the associated birth-death process comprises a finite number of states. For this type of queueing system, the steady-state probabilities are [123, p. 434]

$$p_k = \frac{(\lambda/\mu)^k/k!}{\sum_{i=0}^{c}(\lambda/\mu)^i/i!} \quad \text{for } 0 \leq k \leq c \tag{2.8}$$

If $k = c$, then (2.8) yields the fraction of time that all the servers are occupied, or correspondingly the probability that an arriving patient is lost. In this case, the expression is also known as the *Erlang's loss formula*, which generalizes to any service-time distribution [123, p. 434]. In other words, the $M/G/c/c$ queue. For this reason, the $M/M/c/c$ queue is a fairly robust model that can be employed to evaluate a system directly, or as a surrogate for a much larger model. Through Part II we will demonstrate the robustness of this simple queue, and how the model can be employed in different contexts including both flow evaluation and optimization.

Due to the simple structure of both the $M/M/c$ and $M/M/c/c$ queueing models they are quite "easy" to apply in practice. Furthermore, even though they do not account for a large range of characteristics of a real-life queueing system, they might still yield its general behavior. For instance, the potential of assigning a new physician to a task, or routing more patients to a specific ward, which can be useful in deciding on a scope for a more adequate model. On the other hand, one of the characteristics that define patient flow is the

network of processes that must come together to properly treat the patients. Another characteristic is time-dependency. The amount of staff and the rate at which patients arrive at the hospital might fluctuate over time. Especially in an emergency context.

In the following section, we will elaborate on employing continuous-time Markov chains for modeling patient flow in a network of queues. In addition, we touch upon modeling these networks with time-dependency, which is otherwise elaborated in Chapter 5.

2.1.1 Markov Chains for Modeling Queues

Markov chains are stochastic processes that are based on the notion that a system can be defined by a set of states referred to as the *state space*, S. The process can only attain a single state $s \in S$ at a time, but in return change (or *transition*) between the states as the process evolves. Further, the name *Markov* is derived from the Markovian property of the process, meaning that if at time t_k the process is in state s_k, then a transition at time t_{k+1} to a new state s_{k+1} is only dependent on s_k, and not on the past history of the system. Say we let $X(t) \in S$ define a stochastic process that evolves over time $t \geq 0$, and further that this process attains the states in a sequence $t_0 < t_1 < \cdots < t_{k-1} < t_k < t_{k+1}$, where $k \in \mathbb{N}_0$ is the index of the sequence, then for a Markov chain [123, p. 252]

$$\begin{aligned} Prob\{X(t_{k+1}) = s_{k+1} | X(t_k) = s_k, X(t_{k-1}) = s_{k-1}, \ldots, X(t_0) = s_0\} \\ = Prob\{X(t_{k+1}) = s_{k+1} | X(t_k) = s_k\} \end{aligned}$$

Consider for instance the birth-death process for the aforementioned $M/M/c$ queue in Figure 2.2. The number of patients in the system is either due to a recent arrival or discharge. Furthermore, because the inter-arrival and inter-service times follow a continuous distribution, a transition can occur at any point in time; hence $t \in \mathbb{R}_0$. In this section, we limit our scope to exactly this type of models, namely the Continuous-Time Markov Chains (CTMCs).

The queueing systems we have considered so far are time-homogeneous in terms of their associated CTMC. For the remaining of this section, let $s \in S$ and $s^* \in S$ define the current and a subsequent state in the chain, respectively. Further, let $p(\tau)_{ss^*}$ define the state transition probability of changing from state s to the new state s^* over the time-interval $\tau \in \mathbb{R}_0$. Then, for a homogeneous CTMC the probability $p_{ss^*}(\tau) = Prob\{X(t+\tau) = s^* | X(t) = s\}$ where t can be any point in time. In addition, for any value of τ, the probability is conserved, so $\sum_{s^* \in S} p_{ss^*}(\tau) = 1$. On the contrary, if the CTMC is time-inhomogeneous, then $p_{ss^*}(t, t^*) = Prob\{X(t^*) = s^* | X(t) = s\}$. Thus, the probability of the transition from s to s^* depends on both the time, t, at which point the process is in the current state, s, and the time, t^*, at which point the process has changed to the new state, s^*.

As these state-transitions occur in continuous time it is convenient to define the CTMC based on the *rates* at which the transitions occur instead of their probability. Let $q_{ss^*}(t)$ denote the rate at which the process changes from the current state s to a new state s^* at time t. In the setting of Stewart, 2009 [123, p. 254], this is defined as

$$q_{ss^*}(t) = \lim_{\Delta \to 0} \frac{p_{ss^*}(t, t+\Delta)}{\Delta} \tag{2.9}$$

where $s \neq s^*$, and for homogeneous CTMCs (that are independent of t)

$$q_{ss^*} = \lim_{\Delta \to 0} \frac{p_{ss^*}(\Delta)}{\Delta}$$

once again given that $s \neq s^*$. Now, for the rates where $s = s^*$, namely $q_{ss}(t)$, we have that

$$q_{ss}(t) = -\sum_{s^* \in S \setminus \{s\}} q_{ss^*}(t) \tag{2.10}$$

so the rate associated with the process staying in the current state is in other words constrained by $q_{ss}(t) \leq 0$, where $q_{ss}(t) = 0$ makes s an absorbing state since $q_{ss^*}(t) = 0 \; \forall s^* \in S \setminus \{s\}$. This applies if the CTMC is homogeneous in time as well. Together, $q_{ss^*}(t) \; \forall s, s^* \in S$ make up an $|S| \times |S|$ matrix $Q(t)$, which we refer to as the *transition rate matrix*. On matrix-form we have that

$$Q(t) = \lim_{\Delta \to 0} \frac{P(t, t+\Delta) - I}{\Delta}$$

where $P(t, t+\Delta)$ is the *probability transition matrix* associated with the probabilities $p_{ss^*}(t, t+\Delta)$ and I the identity matrix. Notice that due to (2.10) every row in the transition rate matrix, $Q(t)$, must sum to zero.

Consider the $M/M/c/c$ queue. That is, a queue where arriving patients are lost if all of the c servers are occupied. The arrivals are further generated according to a Poisson process with rate λ, and the inter-service times are exponentially distributed corresponding to a discharge rate of μ. All parameters are independent of time, and the $M/M/c/c$ queue is therefore a homogeneous CTMC with finite state space $S = \{0, 1, \ldots, c-1, c\}$. The transition rate matrix that corresponds to this type of queue is

$$Q = \begin{pmatrix} -\lambda & \lambda & 0 & \cdots & & 0 \\ \mu & -(\mu+\lambda) & \lambda & 0 & \cdots & \\ 0 & 2\mu & -(2\mu+\lambda) & \lambda & & \vdots \\ & & & \ddots & & \\ \vdots & & & & & 0 \\ & & & (c-1)\mu & -((c-1)\mu+\lambda) & \lambda \\ 0 & & \cdots & 0 & c\mu & -c\mu \end{pmatrix}$$

showing that the state (number of patients in the system) increases with a rate of λ, and decreases with a rate of $s\mu$. The diagonal elements, q_{ss}, are therefore $-(s\mu+\lambda)$ to ensure that $\sum_{s^* \in S} q_{ss^*} = 0$ for all states $s \in S$. Naturally, for larger and more complex systems this way of presenting the structure of Q can quickly become immense and therefore confusing. For this reason, we usually present the transition rate matrix on the form

$$q_{ss^*} = \begin{cases} \lambda & \text{if } s^* = s + 1 \\ s\mu & \text{if } s^* = s - 1 \end{cases}$$

where all other transition rates $q_{ss^*} = 0$ for $s \neq s^*$, and $q_{ss} = -\sum_{s^* \in S \setminus \{s\}} q_{ss^*}$ for $s = s^*$. As we will show in the following (and throughout Part II and III of this thesis), the CTMC can be used to model more advanced system characteristics based on the structure of transition rate matrix Q.

Example of a Queueing System

Consider a geriatric nursing ward that always features $n \in \mathbb{N}_{>0}$ inpatients. Furthermore, assume that $m \in \mathbb{N}_{>0}$ nurses are always assigned to this ward of which $u \in \mathbb{N}_0$ nurses are students, and $m - u$ nurses have completed their education and as a result are more experienced. Suppose, that $w \in \mathbb{N}_0$ patients have almost recovered enough to be discharged, whereas $n-w$ patients are in such a bad shape that they can only be attended by the experienced nurses. For this reason, management has decided that the w recovered patients should always be attended by a student nurse; unless none of the students are available, at which point the patients will be attended by an experienced nurse. If none of the experienced nurses are available, we assume for simplicity that the patients do not have to wait, but additional nurses can be summoned from a different nursing ward. An overview of the relations between patients and nurses is depicted in Figure 2.3.

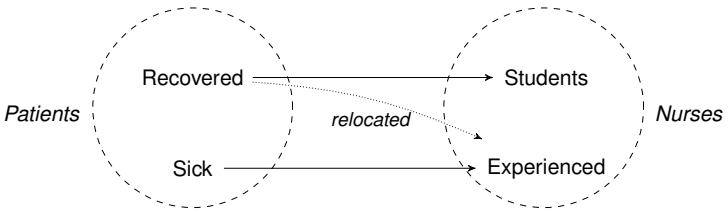


Figure 2.3: Allocation of two types of patients (*recovered* and *sick*) to two types of nurses (*students* and *experienced*).

Let $I = \{recovered, sick\}$ define the set of the two patient types, and $J = \{students, experienced\}$ define the set of the two nurse types, respectively. Then, to sum up the problem, we have:

- A total of $n \in \mathbb{N}_{>0}$ patients of two types, $I = \{recovered, sick\}$.
- $w \in \mathbb{N}_0$ (almost) recovered patients, where $w \leq n$.
- A total of $m \in \mathbb{N}_{>0}$ nurses of two types, $J = \{students, experienced\}$.
- $u \in \mathbb{N}_0$ student nurses, where $u \leq m$.

In order to model the problem as a Markov chain we further require a definition of state $s \in S$, with state space of the system S. Let $x_{ij} \in \mathbb{N}_0$ define the number of patients of type $i \in I$ that are attended by nurses of type $j \in J$. Notice, because we assume that patients do not have to wait for a nurse, x_{ij} only has to account for the patients that are attended. The student nurses are then subject to

$$x_{sick,students} = 0$$

and

$$x_{recovered,students} \leq \min\{w - x_{recovered,experienced}, u\}$$

Correspondingly, for the experienced nurses we have that

$$x_{recovered,experienced} \leq \min\{\max\{w - u, 0\}, \\ m - u - x_{sick,experienced}\}$$

and

$$x_{sick,experienced} \leq \min\{n - w, m - u - x_{recovered,experienced}\}$$

Thus, x_{ij} can be employed to define the state, $s \in S$, of the system at any time. That is,

$$s = (x_{recovered,students}, x_{recovered,experienced}, x_{sick,experienced})$$

which based on the aforementioned constraints yield a state space of size

$$|S| = (\min\{u, w\} + 1)\Bigg(1 + \min\{\max\{w - u, 0\}, m - u\} + \\ \sum_{i=0}^{\min\{\max\{w-u,0\},m-u\}} \min\{n - w, m - u - i\}\Bigg)$$

Assume that the ward contains a total of $n = 25$ patients of which $w = 3$ patients are almost recovered. Furthermore, assume that a total of $m = 5$ nurses are assigned to the ward, and that $u = 1$ of these nurses is a student. Thus, the state space contains a total of $|S| = 24$ states.

Now, let $\lambda_i \in \mathbb{R}_0$ define the rate at which patients of type $i \in I$ require attention from a nurse, and assume that all patients require attention according to a Poisson process. Furthermore, let $\mu_{ij} \in \mathbb{R}_0$ define the service rate for a nurse of type $j \in J$ that attends a patient of type $i \in I$, and assume that inter-service times follow an exponential distribution. Thus, the problem can be modeled as a homogeneous continuous-time Markov chain. Let $q_{ss^*} \in \mathbb{R}$ define the rate at which the system changes from a current state $s \in S$ to a new state $s^* \in S$. Then,

$$q_{ss^*} = \begin{cases} \lambda_{recovered} & \text{if } s^* = (x_{recovered,students} + 1, \dots) \text{ and } x_{recovered,students} < u, \\ & \sum_{j \in J} x_{recovered,j} < w \\ \lambda_{recovered} & \text{if } s^* = (\dots, x_{recovered,experienced} + 1, \dots) \text{ and } x_{recovered,students} = u, \\ & \sum_{i \in I} x_{i,experienced} < m - u, \sum_{j \in J} x_{recovered,j} < w \\ \lambda_{sick} & \text{if } s^* = (\dots, x_{sick,experienced} + 1) \text{ and } \sum_{i \in I} x_{i,experienced} < m - u, \\ & x_{sick,expeienced} < n - w \\ x_{ij}\mu_{ij} & \text{if } s^* = (\dots, x_{ij} - 1, \dots) \text{ and } x_{ij} > 0 \quad \forall i, j \in I, J \end{cases}$$

where all other transition rates $q_{ss^*} = 0$ for $s \neq s^*$, and $q_{ss} = -\sum_{s^* \in S \setminus \{s\}} q_{ss^*}$ for $s = s^*$.

In the above, any expression that follows an "and" defines a constraint that is associated with the rate, q_{ss^*}, and the state-change that has been specified for s^*. Notice that for all of the arrival rates, λ_i, we firstly have to make sure that there is a sufficient number of nurses available. For instance, if the rate $q_{ss^*} = \lambda_{sick}$, then we must make sure that $\sum_{i \in I} x_{i,experienced} < m - u$, where $m - u$ is the total number of experienced nurses and $\sum_{i \in I} x_{i,experienced}$ is the total number of experienced nurses that are currently attending a patient for state $s \in S$. Secondly, since the ward can only take a finite number of both patient types, we have to specify that only this finite number of patients can be attended, and not more. Hence, we add the constraint $x_{sick,expeienced} < n - w$, where $n - w$ is the total number of sick patients, and $x_{sick,expeienced}$ is the number of sick patients that are currently being attended by a nurse.

To illustrate, say $n = 25$, $w = 3$, $m = 5$, and $u = 1$. Then, the transition $s = (0, 0, 2) \to s^* = (\mathbf{1}, 0, 2)$ occurs with a rate of $q_{ss^*} = \lambda_{recovered}$. This is the same for the transition $s = (1, 0, 2) \to s^* = (1, \mathbf{1}, 2)$. However, the transition $s = (0, 0, 2) \to s^* = (0, \mathbf{1}, 2)$ is not allowed, because $x_{recovered,students} = u$ is violated, and the transition therefore occurs with a rate of $q_{ss^*} = 0$. Lastly, $s = (1, 1, 2) \to s^* = (1, 1, \mathbf{3})$ occurs with $q_{ss^*} = \lambda_{sick}$, at which point $m = 5$; hence any further arrivals are not allowed.

Regarding the service rates, the only dependency is the current number of patients of type $i \in I$ that are attended by nurses of type $j \in J$, which has

to be positive. That is, $x_{ij} > 0 \quad \forall i, j \in I, J$. Further, notice that the resulting service rates are governed by the product between μ_{ij} and the number of patients that are currently being attended, leading to $x_{ij}\mu_{ij}$.

Consider the state $s^* = (0, 1, 2)$. Even though, $s = (0, 0, 2) \rightarrow s^* = (0, \mathbf{1}, 2)$ is not allowed, the system can still attain this state through the transition $s = (1, 1, 2) \rightarrow s^* = (\mathbf{0}, 1, 2)$ with a rate of $q_{ss^*} = 1 \cdot \mu_{recovered,students}$.

Probability Distributions

Consider a time-interval of size $t + \Delta$ during which a homogeneous CTMC changes from a state $s \in S$ to a new state $s^* \in S$, and during this transition passes through the state $k \in S$ after time t. Then we have that [123, p. 257]

$$p_{ss^*}(t + \Delta) = \sum_{k \in S} p_{sk}(t)p_{ks^*}(\Delta) = \sum_{k \in S \setminus \{s^*\}} p_{sk}(t)p_{ks^*}(\Delta) + p_{ss^*}(t)p_{s^*s^*}(\Delta)$$

where $t, \Delta \in \mathbb{R}_0$. From the above it can be shown that [123, p. 258]

$$\frac{dp_{ss^*}(t)}{dt} = \sum_{k \in S} p_{sk}(t)q_{ks^*} \quad \forall s, s^* \in S$$

or in matrix form

$$\frac{dP(t)}{dt} = P(t)Q$$

also known as the *Kolmogorov forward equations*. The solution to these differential equations are

$$P(t) = e^{Qt} \tag{2.11}$$

which yields the relationship between probability matrix $P(t)$ of any transition $s \in S$ to $s^* \in S$, transition rate matrix Q, and the interval of observation t [123, p. 258]. Thus, starting from an arbitrary state $s \in S$, the expression (2.11) is useful for describing the evolution of the homogeneous CTMC over time, which may converge into steady-state as $t \rightarrow \infty$. We will elaborate more on the definition of the resulting state probability distribution in what follows.

Assume we observe the progression of a CTMC during time t, and let $\pi_s(t) = Prob\{X(t) = s\}$ be the probability that the system is in state $s \in S$ after time t. Furthermore, let $\pi(t)$ define a vector of size $|S|$ that comprises $\pi_s(t) \; \forall s \in S$. In other words, the probability distribution of the $|S|$ states in the CTMC after time t. For $\pi(t)$ we have that [123, p. 262]

$$\pi(t) = \pi(0)e^{Qt} \tag{2.12}$$

where $\pi(0)$ is the probability distribution at $t = 0$.

Consider a homogeneous CTMC that at $t = 0$ attains a particular state probability distribution $\pi(0)$ — e.g. if the process starts in a specific state, such that $\pi(0) = (1, 0, 0, \ldots, 0)$. Then as time progresses, the process evolves from this state and does so in accordance with (2.12). If the CTMC is not in steady-state at time t, then the process is still *transient*, and the value of $\pi(t)$ changes as function of t. In this case [123, p. 263],

$$\frac{d\pi(t)}{dt} = \pi(t)Q \tag{2.13}$$

If the distribution $\pi(t)$ converges after a sufficient amount of time, such that $d\pi(t)/dt = 0$, then (2.13) can be reduced to

$$\pi Q = 0 \tag{2.14}$$

where π represents the limiting state probability distribution of the CTMC. Notice that $\|\pi\|_1 = 1$. If the CTMC is finite and irreducible, meaning that the process does not have a closed subset of states in S, then the limiting distribution, π, always exists [123, p. 263].

The expression in (2.14) is referred to as the *global balance equations*, and serve as the basis for obtaining the state distribution for a homogeneous CTMC that has attained steady-state.

Solution Approach

The transition rate matrix Q, provides a tool for modeling many advanced characteristics of a queueing system, and by employing this matrix to the system of equations in (2.14), we are theoretically able to solve the steady-state probability distribution. We should emphasize the word *theoretically*, because a queueing system (e.g. a network of queues) of realistic size can easily yield a huge state space, which can be quite computationally expensive for many standard solution methods. Thus, in order to solve (2.14), we propose a numerical approach that derives π within some predefined tolerance limit.

To specify, we consider a problem $f(x) = 0$ that represents a linear system of equations of the type $Ax - b = 0$. An iterative form is then derived by changing this to $x^{k+1} = g(x^k)$, where x^k defines the solution at the k'th iteration. In our case, we consider the system of equations $\pi Q = 0$, cf. (2.14), which after transposing becomes $Q^T\pi^T = 0$. We then employ the recursive formulation $\pi^{k+1} = g(\pi^k)$, with some initial value π^0, until convergence.

Some of the iterative methods for defining the function $g(\pi^k)$ are *Jacobi*, *Gauss-Seidel* and *Successive Over-Relaxation* (SOR) [123, p. 305]. These methods are all characterized by employing the recursive form

$$\pi^{k+1} = H\pi^k$$

where H is referred to as the *iteration matrix*, which is defined differently in each method. In this thesis, we have relied on SOR to derive the steady-state distribution, and for this reason we will elaborate on how to employ this

method in the following.

Consider the system $Q^T\pi^T = 0$, and observe that $Q^T = D - L - U$, where D defines a diagonal matrix, L a strictly lower triangular matrix, and U a strictly upper triangular matrix. For SOR the iteration matrix is defined as

$$H_\omega = (D - \omega L)^{-1}[(1-\omega)D + \omega U] \tag{2.15}$$

where $\omega \in \mathbb{R}$ is a parameter that can be adjusted to control the convergence of the method. In this regard, convergence can only be attained in the range $0 < \omega < 2$. Furthermore, the optimal convergence-rate of SOR is achieved by choosing ω such that the difference between the unit eigenvalue and the subdominant eigenvalue of matrix H_ω is maximized [123, p. 312]. Thus, prior to solving for the probability distribution, π, it might be profitable to conduct a number tests with different values of ω, where H_w is constructed through the expression in (2.15), and subsequently deriving the associated eigenvalues. Literature on the optimal value of ω, given a specific structure of the problem, includes Young, 1954 [145] and Varga, 1959 [133]. Alternatively, a heuristic search procedure can be employed to derive a "good" value for ω. If the CTMC is used in a setup where the values in Q can change, then it might be preferable to conduct a range of experiments with different representative structures of Q to find a value for ω that performs well for most cases. This approach is particularly useful if the performance of SOR is insensitive to the value of ω, and thus, this is the approach that have been employed in this thesis.

Substituting (2.15) with matrix H in $\pi^{k+1} = H\pi^k$, yields

$$\pi^{k+1} = (D - \omega L)^{-1}[(1-\omega)D + \omega U]\pi^k \tag{2.16}$$

which is used to solve for the steady-state distribution, π. Regarding, the application of this approach, the reader may notice that the expression in (2.16) is rather computationally expensive if applied in its current form. Fortunately, that does not have to be the case.

Returning to the general form, $Ax - b = 0$, SOR can be computationally implemented by employing the expression [123, p. 315]

$$x_i^{k+1} = (1-\omega)x_i^k + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij}x_j^k\right)$$

where x_i, b_i and a_{ij} are elements associated with the vectors x and b; and the matrix A, respectively. Further, n is the number of equations as well as unknowns in the system; hence $n = |S|$. We can reduce this expression by firstly recalling that $b_i = 0$ for all i elements, and secondly by scaling the matrix A in each row so the diagonal elements $a_{ii} = 1$. Thus,

$$x_i^{k+1} = (1-\omega)x_i^k - \omega\left(\sum_{j=1}^{i-1} a_{ij}x_j^{k+1} + \sum_{j=i+1}^{n} a_{ij}x_j^k\right)$$

which further leads to [123, p. 316]

$$x_i^{k+1} = x_i^k - \omega \left(\sum_{j=1}^{i-1} a_{ij} x_j^{k+1} + x_i^k + \sum_{j=i+1}^{n} a_{ij} x_j^k \right). \tag{2.17}$$

In our next step, we assume that transition rate matrix $Q \Leftrightarrow Q^T = A$ is sparse such that the matrix can be saved in a much more compact format than what is immediately available from standard matrix-form. In this format, referred to as the Harwell-Boeing format [123, p. 315], we only save the non-zero values of A and does so in a one-dimensional array. Moreover, to keep track of the original location of each element, we further require two arrays that contain information about the row and column indices of each element. Let $\alpha[k] = a_{ij}$ and $\beta[k] = j$ define arrays that contain the value a_{ij} and the corresponding column index j, respectively. Assume that the elements in each of these arrays are sorted in ascending order according to their rows (but not necessarily according to their columns). Furthermore, let $\gamma[i] = k$ define an array of length $n + 1$ that states the position of the first element (in the arrays α and β) for row i. In other words, the non-zero elements of row i are placed in the range $\gamma[i] \leq k < \gamma[i + 1]$, and the number of non-zero elements in this row is $\gamma[i + 1] - \gamma[i]$.

Suppose we employ the Harwell-Boeing format to store each non-zero element of A. Then, expression (2.17) leads to the computationally "friendly" version of SOR, presented in Algorithm 1.

Algorithm 1 A single iteration of successive over-relaxation using the Harwell-Boeing format for matrix A [123, p. 316].

```
1:  x ← initialize()
2:  for i = 1 to n do
3:      sum ← 0
4:      initial ← γ[i]                          ▷ Get range of row elements
5:      last ← γ[i + 1] − 1
6:      for j = initial to last do              ▷ Calculate the sum
7:          sum ← sum + α[j] · x[β[j]]
8:      end for
9:      x[i] ← x[i] − ω · sum                   ▷ Update the i'th element of x
10: end for
        return x
```

Here, the array x is initialized using an arbitrary vector with a 1-norm equal to 1, for instance based on an estimate of the steady-state distribution.

Algorithm 1 shows a *single* iteration of SOR, and thus line 2-10 needs to be run a (possible large) number of times to attain convergence within a predefined tolerance, ϵ. So, the question is how to detect convergence and avoid terminating the recursion too early. Recall that π^k defines the probability

distribution at iteration k, and π the *true* probability distribution. Thus, as $k \to \infty$, $\pi^k \to \pi$.

One approach is to calculate the maximum difference between each successive iteration, and terminate when

$$\max_{s \in S}\{|\pi_s^k - \pi_s^{k-1}|\} < \epsilon$$

whereas, if the elements of π^k are small, this should be replaced with the relative measure

$$\max_{s \in S}\left\{\frac{|\pi_s^k - \pi_s^{k-1}|}{|\pi_s^k|}\right\} < \epsilon \tag{2.18}$$

which is the case for many large state spaces. Notice that because the above methods evaluate *successive* iterates, they are mostly useful if convergence occurs rather fast. Suppose instead that the process converges at a slow rate, such that the change between each iteration is smaller than ϵ, but the true distribution, π, is still far off. In this case, terminating the process prematurely will result in a violation of ϵ if convergence is evaluated over a larger number of iterations. That is, (2.18) should be changed to

$$\max_{s \in S}\left\{\frac{|\pi_s^k - \pi_s^{k-m}|}{|\pi_s^k|}\right\} < \epsilon \tag{2.19}$$

where $m \in \mathbb{N}$. Moreover, notice that (2.19) can substitute (2.18) entirely if parameter m is employed as a function of the convergence rate, such that m starts small, but increases as the process starts to slow down, and more iterations become available [123, p. 318]. The exact definition of such a function would then depend on the specific problem instance. Throughout this thesis, we have determined this function experimentally.

The reader may notice that we have paid particular attention to the instances where the CTMC is homogeneous and in steady-state. However, if the process is transient, so $\pi(t)$ changes as a function of the time of observation t, then recall that the process is governed by

$$\pi(t) = \pi(0)e^{Qt}.$$

The question is how to calculate the matrix exponential for large state spaces. In Chapter 5 we will elaborate on how to achieve this by employing the method of uniformization [61] for a specific hospital problem.

2.1.2 Discrete Event Simulation

The CTMC provides an analytical approach for modeling queueing systems with many real-life characteristics. These include network behavior with probabilistic or state-dependent routing, relocation upon insufficient capacity, time- and state-dependent rates, and so on. Despite the advantages of the CTMC

presented throughout Section 2.1.1, the reader may have noticed that we always require a state space of tangible size. That is, even though we might be able to model a patient flow system mathematically, the computations can be quite intractable — as we will demonstrate throughout Part II and III. A solution to this problem is to reduce and then validate the model by conducting samples from a simulation of the system; *or* merely employing the simulation model as the only evaluator. The latter is quite a popular approach, as shown in our literature reviews (cf. Chapter 1, 3, and 5).

The field of computer simulation covers a wide range of methods including systems with continuous flow, such as system dynamics simulations [122], to systems where definite integrals are evaluated, i.e. Monte Carlo simulations [77]. Notice that even though patients may flow through a queueing system in continuous time, the changes, also denoted *events*, are indivisible. That is, the patient flow that we consider in this study is always interpreted as a sequence of discrete events, where the system transitions from one state to the next. Therefore, this section introduces a type of simulation known as Discrete Event Simulation (DES); a type of simulations that aim at modeling exactly this kind of flow behavior. We have used DES to validate our models throughout Part II and III. Much more elaborate theory and introductions to the field can be found in Allen, 2011 [12] and Stewart, 2009 [123]. Additionally, more general overviews have been provided by Schriber et al., 2015 [112] and Sanchez, 2007 [110]

A DES is characterized by a sequence of discrete events that govern the entire behavior of the system throughout the simulation period. For a queueing system, these events will occur in continuous time, and upon their occurrence, update the system, possibly followed by a scheduling of new events. As regards a computer implementation of a DES, the exact structure may vary between cases due to the differences in problem structure. However, an implementation of DES will comprise the following two phases:

- An Entity Movement Phase (EMP)
- A Clock Update Phase (CUP)

In general, the program will loop between these phases until a stopping criteria is met, for instance after a pre-defined number of entities have been processed, or when the simulated time exceeds a certain limit. During the EMP an event is executed which yields a change of the system (usually the movement of an entity). Upon completion of the event, the simulation "clock" is then advanced for the next EMP. This *update* is referred to as the CUP [112].

In this thesis, the simulated entities are patients, and thus, in an EMP the entities are moved between a range of nodes that simulate either a queue or a schedule. Further, the CUP must ensure that the time between successive events reflects the delays in the system that are governed by the distribution of service- and inter-arrival times.

During the EMP, entities will be subject to a range of different states. Again, the range and nature of these states, will be greatly dependent on the system that is simulated. Even so, most implementations of DES comprise the following five categories [112]:

1. **Active**

   As long as an entity is being altered (e.g. when a patient is moved from one queue to the next), the entity is in an *active* state. Only one entity can be altered at a time, and upon completion the entity will change to one of the following four *inactive* states.

2. **Ready**

   Even though the simulation is only able to change one entity at a time, there may be a range of entities that are neither waiting nor being processed, but simply in a state where they are *ready* to become active.

3. **Time-delayed**

   At some point during their life-time, entities will have to become deliberately delayed within a randomly sampled amount of time to simulate that the entity is being processed. In simulations of hospital patients flow, this will usually correspond to the time during which a patient is treated by a physician, triaged by a nurse, or occupying an operating room.

4. **Condition-delayed**

   Some entities might be delayed due to some other condition in the model. That is, instead of employing a pre-defined delay, the entity might have to wait for a condition in the model to change. For instance, this state occurs if the simulation has to account for physical entities that are queued due to insufficient capacity, which in hospital patient flow corresponds to the patients waiting for a resource to become available.

5. **Dormant**

   Besides the pre-defined time-delays, or conditional delays, entities might be delayed because they cannot become ready until some user-provided logic has changed. If that is the case, the entities are in a *dormant* state. Thus, dormant entities are closely related to the condition-delayed entities because they have to wait for a specific update in the system. However, the entities cannot be automatically transfered from this state upon a change in the model conditions.

The simulations conducted throughout this thesis do not generally consider all five entity states at the same time. For instance, the simulation conducted in Chapter 3 requires entity states of type 1-3 and chapter 5 requires entity states of type 1-4.

In order to organize the entities as they move through the simulation, we require a storing of the entities on a number of lists, depending on their respective states. These event-lists are used to track and manage the entities throughout their entire life-time. A DES will typically comprise the following four types of lists [112]:

1. **Current Events**

   The list of current events include any entity that has attained the *ready* state. Thus, during the EMP the simulation will run through each of the elements on this list making sure that each of the ready entities are treated by altering their state or removing them from the system.

2. **Future Events**

   Entities that are deliberately delayed (e.g. due to treatment) are transfered to a dedicated list. Here, any known future event in the simulation is stored. Besides tracking the entities, this list must further include the exact points in time at which the entities will end their delay. Thus, when an entity is moved from a time-delayed to a ready state, the CUP uses the time-stamps on this list to advance the "clock" of the simulation. Because this list depicts much of the future progression of the simulation, it is convenient to always keep the elements sorted according to their time of event.

3. **Delays**

   Similar to the future events list, any entity that has been delayed due to the model conditions is stored on a separate list. In order to attain a proper tracking of the entities, several lists might be needed for this type of delay, and all of these lists are maintained automatically by the model. Say we want to simulate the flow of patients through an emergency room. The currently treated patients will be stored on the aforementioned future events list, whereas all of the waiting patients are tracked on a delay list. Here, they will stay until a physician becomes idle, and when this happens, one of the patient entities (depending on the queueing discipline) will be transferred to the current events list, and then later move on to the future events list to simulate that the patient is under treatment.

4. **User-Managed**

   Lastly, a DES may contain a number of lists storing the entities that have been put into the dormant state. Since the model cannot transfer these entities automatically (cf. the definition of the dormant state), these lists must be managed by the user, meaning that the model requires a user-provided logic in order to transfer entities to and from these lists.

There exists a number of commercial softwares which provide the basis for simulating complex systems by modeling the systems as DESs. A few examples of these are: *Arena* by Rockwell Automation [1], and *Plant Simulation* by Siemens [5]. Even so, in order to gain sufficient insight into the progression

of the simulation it is sometimes useful to implement the simulation in a programming language, such as Java or Matlab, which is the approach that we have used in this thesis. In the following, we will present a few examples of our approach.

Time-dependent Arrivals

Consider a simulation of a queueing system, which is open to new arrivals, and that the simulation stops when an event occurs after a pre-defined amount of time. If the arrival process is known, stationary and independent of the specific condition of the system, then we can generate all the arrivals that the simulation will ever need in advance, and add them to the aforementioned future events list. In practice, we employ a function that conducts pseudo-random samples from a distribution that corresponds to the arrival process. Suppose arrivals are generated by a Poisson process, then we require a function that conducts samples from the associated exponential distribution (cf. beginning of Section 2.1). For a Matlab implementation, this would be `exprnd()`. The program can now generate arrivals, as presented in Algorithm 2, by declaring a time-variable, say t, and perform a loop where samples from the pseudo-random function are added to t until the simulation time-limit is exceeded. If the queueing system must account for different entity types (e.g. patients of different classes, such as *recovered*, *sick* and *acute*) that have different inter-arrival times, then successive runs can be conducted where the input parameter for the pseudo-random function is changed correspondingly. The patients must then be sorted in accordance with their time of arrival.

Algorithm 2 A Poisson process arrival-generator (with mean inter-arrival time $1/\lambda$) for a single class of entities.

1: $L \leftarrow \emptyset$ ▷ Initialize
2: $t \leftarrow 0$
3: **while** $t < T$ **do** ▷ Run until time-limit T is exceeded
4: $\quad \delta \leftarrow sampleExponential(1/\lambda)$
5: $\quad t \leftarrow t + \delta$
6: $\quad L \leftarrow add(L, t)$ ▷ Add new arrival to the list
7: **end while**
return L

Arrivals can be generated quite easily as long as the process is stationary by using the above approach. Assume that the arrival process is time-dependent such that the rate, $\lambda(\tau)$, at which entities arrive to the system is a function of time, $\tau \in \mathcal{T}$, where $\mathcal{T}$ can be cyclically ordered. To clarify, consider the arrival of patients to an emergency department. This process can be highly time-dependent (cf. Chapter 5), and governed by a rate that increases during the day and then starts to decrease around late afternoon, following a weekly cyclical pattern.

In our simulation, we accommodate this behavior by employing the same structure as previously by generating entities at a rate corresponding to $\max_{\tau \in \mathcal{T}}\{\lambda(\tau)\}$. However, instead of adding every new entity to the list, we accept entities that are generated at time τ with a probability of $p = \lambda(\tau)/\max_{\tau \in \mathcal{T}}\{\lambda(\tau)\}$. As a result, the arrival rate governs the probability of accepting the entities to the list, resulting in fewer arrivals when the arrival rate is low, and correspondingly more arrivals when the arrival rate is high.

Extending the structure of Algorithm 2 with time-dependent arrivals, we get the arrival-generator presented in Algorithm 3. Notice that we account for a cyclical pattern using the variable τ, and the overall length of the simulation using the variable t. Moreover, since τ can be used to model a cyclical pattern, we must employ a function, $adjust()$, to track when to reset τ in accordance with $\mathcal{T}$. Otherwise, τ is updated similar to t.

Algorithm 3 Extension of Algorithm 2 with time-dependent arrivals.

$L \leftarrow \emptyset$ ▷ Initialize
$t \leftarrow 0$
$\tau \leftarrow 0$
$y \leftarrow max(\lambda(\tau))$ ▷ Maximum arrival rate for all τ
while $t < T$ **do**
  $\delta \leftarrow sampleExponential(1/y)$
  $\tau \leftarrow adjust(\tau, \delta, \mathcal{T})$ ▷ Advance the cyclical time
  $t \leftarrow t + \delta$ ▷ Advance the simulation time
  $r \leftarrow sample(0, 1)$
  $p \leftarrow \lambda(\tau)/y$
  if $r \leq p$ **then** ▷ Accept with probability p
    $L \leftarrow add(L, t)$
  end if
end while
  return L

In Chapter 5, the method presented in Algorithm 3 has been used to generate arrivals in a DES of an emergency department. In this simulation, we account for time-dependent arrivals of multiple classes that occur according to a Poisson process following a weekly cyclical pattern.

Example of a DES Queueing System

Consider a simulation of an Emergency Department (ED) for which all future arriving patients have been generated by employing Algorithm 3. In addition, assume that the ED triage their patients and that each triage-level occurs with an independent pattern. To accommodate this, Algorithm 3 has been run a successive number of times, where the arrival function has been changed for each triage-level. The resulting list is then sorted such that arrivals occur in ascending order.

Now, assume that the arriving patients can only be triaged by a specialized nurse. Upon completion of the triage, patients are transferred to a physician for examination, after which they may be discharged, or re-triaged and treated by a specialized physician. There is a limited number of each staff type available which fluctuate in accordance with a weekly working-pattern, similar to the arrival rate.

This system may be modeled as a DES by employing Algorithm 4. In this implementation, all known events are stored on the list L along with information about the type of each event. Furthermore, a list, Q, is used to store all of the entities that are waiting for an idle member of the staff. For simplicity, Algorithm 4 focuses on the patient flow, and *does not* show the simultaneous tracking of staff capacity.

The simulation initializes by resetting the simulation "clock", t, and queues in the system Q. Further, all patients that are needed throughout the simulation period are generated in advance and stored in L. The simulation then loops by selecting the first event in L, where the event type is checked, along with the current simulation time, t. Next, a series of actions are conducted depending on whether the event is an *arrival*, *routing*, or a complete *discharge* from the system. If the event is either an arrival or routing, then the patient will require attendance from a member of the staff. The simulation must therefore decide on an appropriate staff type for the patient, and whether this particular type is idle at the moment. This is handled in line 8. If a member of the staff is idle, then an event, specifying when the service (e.g. treatment) is finished, is added to L (cf. line 9); otherwise the patient is added to the queue associated with the staff type in Q (cf. line 11). Here, the queueing discipline is respected by sorting patients according to their time of arrival and respective triage-level.

In addition to moving a patient, the simulation has to make sure that patients are transferred from queue to service when a staff member becomes idle, which is the case if the recent event is either a routing or discharge. The simulation checks this in line 14, after which line 15 checks for any waiting patients in the queue associated with the idle staff member.

Lastly, notice that if the event is a discharge, the simulation does not add a new event to L, but merely changes a staff member to idle (which is not shown in Algorithm 4), and then checks the associated queue for any waiting patients.

2.2 Heuristic Optimization

In the previous section we elaborated on a number of mathematical and computational tools that can be employed to model the behavior of patient flow. Hospital management may wish to exploit these tools in order to evaluate the effect of procuring or reallocating resources. However, suppose that a hospital department is subject to a budget cut, but still needs to increase performance in order to attain the department targets. Targets that may even be set by

Algorithm 4 An example of a DES implementation for an emergency department.

$t \leftarrow 0$
$Q \leftarrow \emptyset$
$L \leftarrow generateArrivals(T)$ ▷ Pre-generate all arrivals
while $t < T$ **do**
    $eType \leftarrow getType(L[1])$ ▷ Get type of the event
    $t \leftarrow getTime(L[1])$

    if $isArrival(eType)$ or $isRouting(eType)$ **then** ▷ Inflow of patients
        if $staffIdle(eType)$ **then**
            $L \leftarrow addServiceEvent(t, eType, L)$
        else
            $Q \leftarrow addToQueue(eType, Q)$
        end if
    end if

    if $isRouting(eType)$ or $isDischarge(eType)$ **then** ▷ Update queue
        if $notEmpty(eType, Q)$ **then**
            $Q \leftarrow removeFromQueue(eType, Q)$
            $L \leftarrow addServiceEvent(t, eType, L)$ ▷ From queue to service
        end if
    end if

    $L \leftarrow removeEvent(L[1])$
    $L \leftarrow sort(L)$
end while

the government. Then, even though the department may be able to model its patient flow mathematically, such queueing model would not provide an immediate answer as to how resources should be re-configured so performance is maximized without violating any budget constraints. Furthermore, suppose the queueing model has a structure, such that the exact optimum cannot be derived without complete enumeration. Then, one approach might be to test a number of manually designed solutions, evaluate their effect, and then simply apply the best one. Another approach would be to apply an algorithm that automatically searches partial areas of the feasible solution space, and returns the best known solution to the decision maker. This type of algorithm is also referred to as a *heuristic* [75], whereas in this thesis, we will often use the convention *heuristic search procedure*, or merely *search procedure*, to distinguish these algorithms from the other heuristic methods that we apply.

In the following we will briefly present the basic concepts of a heuristic search procedure, and present some of the well-known algorithms. Specifically, we will focus on a branch of algorithms known as *local* search procedures that will be widely applied to optimize patient flow throughout this thesis. Lastly, we will elaborate on some algorithm structures that we have found particular useful for evaluating patient flow based on the methods in Chapter 2.1.

Once again, consider the aforementioned hospital optimization problem. In order to conduct any form of optimization, we require a tangible measure of the performance that the department management aims to optimize. Let $g(\mathbf{x})$ define a function that yield this measure, e.g. the expected number of patients that can be examined by a physician per day, where $\mathbf{x}$ is a vector that governs the performance of the system. Let the set I account for all variables that affect the performance of the system such that $|I|$ is the size of $\mathbf{x}$ and the i'th element, x_i, are defined by $i \in I$. Furthermore, let $f_j(\mathbf{x})$ define a function that induces a cost of type $j \in J$, where J is a set that constitutes all cost types. Lastly, let b_j define a budget constraint of type $j \in J$. The optimization problem can then be stated on the form

$$\text{Maximize} \qquad g(\mathbf{x}) \tag{2.20}$$

$$\text{Subject to} \quad f_j(\mathbf{x}) \leq b_j \quad \forall j \in J \tag{2.21}$$

$$\mathbf{x} \in \mathbb{N}_0 \tag{2.22}$$

which represents the most general form of the problem. We have included constraint (2.22), since nonnegative integers are often a requirement [75]. Otherwise, the vector $\mathbf{x}$ can contain elements that are real, or a mixture of both real and integer.

If the optimization problem can be formulated as a system of linear inequalities, we can change (2.20)-(2.22) to the form

$$\text{Maximize} \quad \mathbf{cx} \tag{2.23}$$

$$\text{Subject to} \quad \mathbf{Ax} \leq \mathbf{b} \tag{2.24}$$

$$\mathbf{x} \in \mathbb{N}_0 \tag{2.25}$$

where $\mathbf{c}$ is a vector of coefficients of size $|I|$, $\mathbf{b}$ is a vector of size $|J|$ containing element b_j for all $j \in J$, and $\mathbf{A}$ is a matrix of size $|J| \times |I|$ of coefficients that relate to the $|I|$ problem variables and $|J|$ cost types. However, the linear form is inadequate if the patient flow is evaluated by directly employing the methods from Section 2.1.1. Therefore, throughout Part II and III we often require a combined form. For instance,

$$\text{Maximize} \quad g(\mathbf{x}) \tag{2.26}$$

$$\text{Subject to} \quad \mathbf{Ax} \leq \mathbf{b} \tag{2.27}$$

$$\mathbf{x} \in \mathbb{N}_0 \tag{2.28}$$

where $g(\mathbf{x})$ is a function that cannot be expressed linearly.

2.2.1 Fundamental Heuristic Search Procedures

From this point we will refer to the vector $\mathbf{x}$ as the *solution* to the optimization problem. Further, by letting polyhedron X define the space of feasible solutions, according to (2.27) and (2.28), we shall refer to X as the *search space*. A heuristic search procedure is defined by a *strategy* that selects and evaluates different solutions, $\mathbf{x}$, constraint by X, based on the objective function $g(\mathbf{x})$. The strategy does not have to rely on a mathematical model, other than the model required to calculate the objective value. However, if that is the case, for instance if sub-optimal conditions are exploited by solving a part of the problem using integer linear programing, we refer to the algorithm as a *matheuristic* search procedure.

Similar to DES (cf. Section 2.1.2), a successful implementation of a heuristic search procedure depends very much on the given problem. However, there is a number of general *metaheuristic* structures that can be exploited to form an efficient heuristic for a wide variety of problems. In the following, we will present the metaheuristic structures that have been employed throughout this thesis:

Hill Climber Consider the optimization problem in (2.26)-(2.28) and let $\mathbf{x}^k$ define a feasible solution to the problem at iteration $k \in \mathbb{N}$. Further, let $\mathcal{N}(\mathbf{x}^k)$ yield a set of "neighboring" potential replacements to $\mathbf{x}^k$, also known as the *neighborhood* to $\mathbf{x}^k$. In a hill climber heuristic, each iteration is initialized by

considering the current neighborhood. Solutions are recursively collected from this neighborhood and evaluated using $g(\mathbf{x}^k)$. If a neighboring solution yields a *better* (i.e. for the current problem, higher) objective value than $\mathbf{x}^k$, then the next iteration commences by using this particular solution as basis for a new neighborhood, $\mathcal{N}(\mathbf{x}^{k+1})$, which the algorithm then starts to evaluate in a similar fashion [75, p. 9]. The notion of the hill climber is to keep introducing small changes until the objective value stops improving, at which point the algorithm is terminated. Otherwise, the algorithm can be terminated if the elapsed time exceeds a pre-defined limit.

The exact definition of the neighborhood depends greatly on the problem at hand, and there may even be several equally suitable definitions for the same problem, which may have to be determined experimentally. Furthermore, as regards a strategy for choosing a solution from the neighborhood, several approaches exist. If the neighborhood is large, it may be beneficial to sort the solutions in random order, and then move to a new solution, $\mathbf{x}^{k+1}$, on a *first-best* basis. On the other hand, for small neighborhoods the overall best solution can be obtained by conducting a complete enumeration of $\mathcal{N}(\mathbf{x}^k)$.

In general, the hill climber results in an intensive search relative to the initial solution, $\mathbf{x}^0$. However, since the hill climber is forced to terminate as soon as the first local optimum is discovered, the algorithm lacks substantial diversification.

Tabu Search The metaheuristic known as *tabu search* was originally proposed by Glover, 1986 [57], and has been applied in a substantial number of papers since then [75, p. 243]. In tabu search, the problem of becoming stuck in a local optimum is solved by allowing the algorithm to keep evaluating new solutions, even if the best known solution cannot be updated in every iteration. Tabu search does this by distinguishing between the *global* best known solution, $\mathbf{x}^*$, and the *current* best solution $\arg\max_{\mathbf{x}' \in \tilde{\mathcal{N}}(\mathbf{x}^k)}\{g(\mathbf{x}')\}$, where $\tilde{\mathcal{N}}(\mathbf{x}^k)$ is defined as an admissible subset of the neighborhood $\mathcal{N}(\mathbf{x}^k)$. To clarify, the algorithm must choose the best solution that it can currently find, but to prevent the algorithm from cycling, certain attributes are marked *tabu*, such that the algorithm does not immediately revisit any currently known solutions. Thus, in the beginning of each iteration, the set of solutions that the algorithm is allowed to evaluate has become reduced, leading to the set $\tilde{\mathcal{N}}(\mathbf{x}^k)$. Hence, we have that $\tilde{\mathcal{N}}(\mathbf{x}^k) \subset \mathcal{N}(\mathbf{x}^k)$.

All prohibited attributes are stored in a list, referred to as the *tabu list*, $\mathcal{T}$. In the most basic implementation of tabu search, the entries are sorted according to their order of occurrence, and the oldest entry is removed whenever the number of elements in $\mathcal{T}$ exceeds a pre-defined limit. In each iteration, the algorithm defines $\tilde{\mathcal{N}}(\mathbf{x}^k)$ based on $\mathcal{T}$, and then conducts a complete enumeration to derive a new solution $\mathbf{x}^{k+1}$. If $\mathbf{x}^{k+1}$ yields a better objective value than $\mathbf{x}^*$, the solution is stored as the new global best known solution. The algorithm then updates the tabu list, and continues the recursion until the termination criteria is satisfied.

To overcome the computational challenges of performing a complete enu-

meration of large neighborhoods, it can be beneficial to employ a method that only evaluates the most promising elements, such as a candidate list [75, p. 174].

Similar to the hill climber, tabu search generally yields an intensive search, but the algorithm is simultaneously able to escape from local optima and therefore less dependent on the choice of the initial solution, $\mathbf{x}^0$.

GRASP The reader may have noticed that neither the hill climber nor tabu search accounts for the choice of the initial solution, $\mathbf{x}^0$. Depending on how the neighborhood as well as the other algorithm elements are defined, the initial solution may have a great impact on the performance of the heuristic. An approach to this problem can be to store the best known solution, and simply restart the heuristic several times during runtime by using different values of $\mathbf{x}^0$. Obviously, a convenient approach is to generate $\mathbf{x}^0$ for each new run of the heuristic automatically. This type of nested heuristic structure is exploited in the Greedy Randomized Adaptive Search Procedure (GRASP) by recursively conducting local search, and initializing each run with a *greedy randomized* solution. This approach was originally proposed by Feo & Resende, 1989 [54].

The greedy randomized solution is constructed by defining and ranking a number of candidates to a solution. That is, suppose we are faced with the problem of assigning a fixed number of physicians to a fixed number of time-slots in a schedule. In the initial solution to this problem, $\mathbf{x}^0$, each physician is assigned to a specific time-slot. To achieve this, a greedy approach would be to loop through and determine the largest contribution of each physician to the objective value, then rank them according to their contribution, and lastly assign each of them in descending order, making sure to recalculate the rank after each assignment. This would immediately yield a deterministic initial solution, which the algorithm overcomes by performing a random selection from a subset of the most promising candidates on the list, which is also known as the *restricted candidate list*.

After the greedy randomized solution has been constructed, the search can be intensified by employing the local search procedure, e.g. a hill climber or tabu search heuristic. For this process, the algorithm requires a termination criteria to ensure that the process is restarted a sufficient number of times, resulting in a thorough exploration of the most promising regions of the search space. Several criteria may be useful in this regard. For instance, an upper limit on the elapsed time or iterations; or whenever a number of iterations have been conducted without any improvement to the best known solution. For the criteria that terminates the GRASP completely, an upper limit on time is usually beneficial.

Due to the balance between intensification and diversification, the GRASP can be quite useful as a basis for many different types of heuristics. That said, in order to succeed in making an efficient implementation of GRASP, one needs to make quite a few decisions as regards termination criteria, structure of the local search procedure, and setting of all parameters that are related

hereto.

More elaborate descriptions of various useful metaheuristics are presented in Kendall & Burke, 2005 [75].

2.2.2 Optimization of Queues with Integer Programming

The aforementioned metaheuristics provide an excellent basis for deriving solutions of high quality due to their inherent adaptability. Nonetheless, in optimization of queueing systems there are situations where the problem can be exploited to form heuristics beyond these fundamental algorithm structures. As we have shown in our literature reviews, cf. Chapter 1, 3 and 5, the literature on optimization of patient flow using queueing theory is far from abundant. However, the amount becomes more considerable if we include other application areas as well. See for instance the survey by Bitran & Morabity, 1996 [24] on optimization of manufacturing systems, or more recently Andriansyah et al., 2010 [14] on optimization of open zero-buffer multi-server queueing networks, and Randhawa, 2016 [105] on optimality gaps.

In the following we will focus on problems that relate directly to optimization of patient flow, and describe how we have exploited these problem structures to derive near-optimal heuristic solutions based on matheuristic models. The aim in this section, is to prepare the reader for the subsequent chapters as well as to demonstrate that these methods are generalizable.

Optimization with Waiting Time Constraints

Once again, suppose a hospital department is subject to budget-cuts and therefore have decided to minimize their expenditures by investigating the potential for reducing the amount of staff. Assume that the department's total monthly cost is a linear function of the employed staff types, I, such that the objective function can be expressed on the form $\mathbf{cx}$, similar to (2.23), where $\mathbf{c}$ defines the monthly cost and $\mathbf{x}$ the number of each staff type, respectively. Minimizing this function will ultimately lead to an increased patient waiting time, since fewer resources will be available to serve the patients. For this reason, the department have identified a set of nodes in the patient's path, denoted J, where the waiting time should not exceed a certain limit for a fraction of the patients. The department has additionally identified a relation between the number of each employed staff type and the fraction of patients that exceed this limit, which can be modeled by a function $W_j(\mathbf{x})$ for all $j \in J$. Additionally, function $W_j(\mathbf{x})$ is constrained by a lower bound denoted b_j for all $j \in J$. Further, assume that all staff types are subject to a number of departmental rules, union settlements, and practical limitations which can be modeled by the system of linear inequalities $\mathbf{Ax} \geq \boldsymbol{\beta}$, where $\boldsymbol{\beta}$ is a vector of length $|K|$, $\mathbf{A}$ is a $|K| \times |I|$ matrix, and K is a set that comprise all necessary staff-constraints. For the convenience of this example, we assume that these constraints can be expressed by employing $\mathbf{x}$ directly without introducing any further dimensions to the vector, hereby finally yielding the optimization problem

$$\text{Minimize} \quad \mathbf{cx} \tag{2.29}$$

$$\text{Subject to} \quad W_j(\mathbf{x}) \geq b_j \quad \forall j \in J \tag{2.30}$$

$$\mathbf{Ax} \geq \boldsymbol{\beta} \tag{2.31}$$

$$\mathbf{x} \in \mathbb{N}_0 \tag{2.32}$$

Due to the complexity of the waiting time functions, $W_j(\mathbf{x})\ j \in J$, we assume that optimality cannot be proven for (2.29)-(2.32) by employing any known solution approach.

Any of the aforementioned metaheuristics would be applicable to this problem. However, assuming that $\mathbf{Ax} \geq \boldsymbol{\beta}$ is solvable by any commercial linear solver, and we can replace the intractable constraints in (2.30) with a "good" linear estimate; then a near-optimal heuristic solution can be derived by solving the resulting system of linear inequalities by using the commercial linear solver.

In order to derive the estimate for (2.30) several approaches might be useful. In this thesis, we use a recursive formulation, where the linear model is solved, followed by an adjustment of the estimate based on an evaluation of the waiting time functions $W_j(\mathbf{x})\ j \in J$. Let Y^k define the estimated space at iteration $k \in \mathbb{N}_0$. The aim is then to choose an initialization where $X \subset Y^0$, and to have Y^k approach X during the recursion. We elaborate more on this modeling approach in Chapter 5.

Room Allocation with Sub-Optimality

In this example, we consider an optimization problem where the objective function attains a complexity such that optimality cannot be proven, similar to the aforementioned case. On the other hand, all constraints that are related to the problem can be expressed by using a system of linear inequalities. As we will show in Chapter 4, this structure applies to the problem of allocating room types among a set of nursing wards.

That is, assume a hospital setting where a fraction of the arriving patients prefer admission to a private room, and the remaining patients have no preference as to whether they are admitted to a private or shared room. Moreover, assume that each of the hospital nursing wards are subject to a limited bed capacity, such that if the capacity is depleted, arriving patients will be relocated to an alternative nursing ward. Notice that even though patients might be relocated, they still maintain their preferences. In order to ensure a high level of service, hospital management has decided to maximize the expected matching between patients and their preferred room type by altering the configuration of the currently available room types. For the convenience of this example, we assume that the resulting number of relocated patients can be disregarded.

Let $g(\mathbf{x})$ yield the total expected number of patient matches as function of room configuration $\mathbf{x}$. Here, x_i is an element to the vector $\mathbf{x}$ and $i \in I$, where $I = \mathcal{R} \times \mathcal{W}$. In addition, $\mathcal{R}$ is the set of all room types, and $\mathcal{W}$ is the set of all wards. Moreover, let $\mathbf{Ax} \leq \mathbf{b}$ comprise the constraints necessary to ensure at least one bed per ward, and that the number of available room types remains fixed, resulting in the optimization problem

$$\text{Maximize} \quad g(\mathbf{x}) \tag{2.33}$$

$$\text{Subject to} \quad \mathbf{Ax} \leq \mathbf{b} \tag{2.34}$$

$$\mathbf{x} \in \mathbb{N}_0 \tag{2.35}$$

Just as previously, any of the aforementioned metaheuristics may be applied to (2.33)-(2.35) by searching through the room configurations, $\mathbf{x}$. However, suppose that $g(\mathbf{x})$ can be evaluated by using the following steps:

1. Determine the total amount of patients arriving to each ward (including relocations) by employing a function $H(\mathbf{x})$, which results in a bed occupancy probability distribution π_w for each ward $w \in \mathcal{W}$.
2. Employ the aforementioned distributions to derive the expected number of private room matches for each ward $w \in \mathcal{W}$ through the function $h_w(\pi_w, \mathbf{x})$.

As previously described, a relocation occurs whenever the capacity of a ward is depleted, regardless of the specific room types. In other words, $H(\mathbf{x})$ can be evaluated using only the aggregated ward capacity. Thus, if $h_w(\pi_w, \mathbf{x})$ can be expressed as a formula that can be evaluated in negligible time compared to $H(\mathbf{x})$, then we may enumerate $h_w(\pi_w, \mathbf{x})$ for each potential number of private rooms $j \in \mathcal{J}_w$ that ward $w \in \mathcal{W}$ can receive. Here, $\mathcal{J}_w = \{0, 1, \ldots, M_w\}$, and M_w is the aggregated capacity of ward $w \in \mathcal{W}$. Let the coefficients $c_{wj}\ \forall w, j \in \mathcal{W}, \mathcal{J}_w$ yield the result of this enumeration. Then, the optimal configuration of rooms can be solved by maximizing $\sum_{w \in \mathcal{W}} h_w(\pi_w, \mathbf{x})$ and employing the linear form

$$\text{Maximize} \quad \sum_{w \in \mathcal{W}} \sum_{j \in \mathcal{J}_w} c_{wj} z_{wj} \tag{2.36}$$

$$\text{Subject to} \quad \mathbf{A}'\mathbf{z} \leq \mathbf{b}' \tag{2.37}$$

$$\mathbf{z} \in \{0, 1\} \tag{2.38}$$

where $z_{wj} = 1$ if j private rooms are assigned to ward w, and $\mathbf{z}$ is a vector that constitutes the element $z_{wj}\ \forall w, j \in \mathcal{W}, \mathcal{J}_w$. Moreover, $\mathbf{A}'\mathbf{z} \leq \mathbf{b}'$ extends the constraints of $\mathbf{Ax} \leq \mathbf{b}$ by ensuring that the aggregated capacity is constant, and that $\sum_{j \in \mathcal{J}_w} z_{wj} = 1\ \forall w \in \mathcal{W}$. Further, we assume that (2.36)-(2.38) is tractable and can be solved using standard theory, or a commercial linear

solver. Thus, by solving the model formulation in (2.36)-(2.38) to optimality, we can be content with designing a heuristic search procedure for the aggregated capacity, which essentially reduces the size of the search space. We elaborate more on this modeling approach in Chapter 4.

2.3 Concluding Remarks

Throughout this chapter we have provided a basis for modeling and optimizing advanced patient flow systems. In order to achieve this, we have provided theory within two general areas: Firstly, the mathematical and numerical methods that are necessary to understand and to evaluate the performance of hospital patient flow. In particular, we have focused on systems that have attained steady-state and how to model these based on theory in Markov chain modeling. In this regard, we have elaborated on how to derive the state probability distribution of the system, but left the calculations of the specific performance measures for the more case-specific chapters. Secondly, we have briefly described a variety of meta- and matheuristic methods that are useful for deriving good heuristic solutions to certain optimization problems. The methods for modeling patient flow can be employed in these optimization algorithms to serve as both constraints and objective functions.

In this regard, the reader should notice that both Markov chains and simulation-based methods are applicable in a heuristic optimization scheme. However, the two approaches comprise very different characteristics which should be accounted for when the search procedure is developed. The most important difference is probably that the output from a DES is random, and for this reason the input into the search procedure, whether it is a constraint or objective function, will be random as well. Thus, if variability is not accounted for, we might accept the bad solutions falsely, and correspondingly some of the good solutions might be rejected. The same goes for the solution feasibility. In addition, since simulation does not provide an analytical understanding of the system, solving the optimization problem to proven optimality is quite difficult, if not entirely ruled out.

Returning to the notion of conducting a heuristic search by employing the methods from Section 2.1. Here, any required information about the system will become immediately available to the search procedure, without the need for sampling. Even though simulation provides a basis for obtaining sufficient samples to attain high accuracy, the necessary runtime can (depending on the specific case and computational implementation) exceed that of the associated CTMC calculations.

Simulation provides a basis for modeling a wide range of advanced characteristics, and is sometimes the only available (adequate) approach for evaluating a flow system. In addition, commercial simulation software has made DES an accessible modeling approach to users that are not familiar with the underlying theory. As a result DES is being used in many application areas.

However, note that from a research perspective, simulations generally provide a poor basis for reproducing results, and that sufficient data may not always be available to account for all attributes of a complex simulation model.

Research into analytical methods is greatly beneficial in the context of optimization, among other things due to the presence of an analytical understanding of the system, and a solid basis for reproducing results. In addition, as the computational capacity gradually increases, so does the relevance and potential of Markov chains that feature large state spaces, which ultimately leads to greater complexities and more realistic system evaluations.

Part II

Inpatient Flow

Chapter 3

Optimization of Hospital Ward Resources with Patient Relocation using Markov Chain Modeling[1]

Anders Reenberg Andersen, Bo Friis Nielsen
and Line Blander Reinhardt

Abstract Overcrowding of hospital wards is a well known and often revisited problem in the literature, yet it appears in many different variations. In this study, we present a mathematical model to solve the problem of ensuring sufficient beds to hospital wards by re-distributing beds that are already available to the hospital. Patient flow is modeled using a homogeneous continuous-time Markov chain and optimization is conducted using a local search heuristic. Our model accounts for patient relocation, which has not been done analytically in literature with similar scope. The study objective is to ensure that patient occupancy is reflected by our Markov chain model, and that a local optimum can be derived within a reasonable runtime.

Using a Danish hospital as our case study, the Markov chain model is statistically found to reflect occupancy of hospital beds by patients as a function of how hospital beds are distributed. Furthermore, our heuristic is found to efficiently derive the optimal solution. Applying our model to the hospital case, we found that relocation of daily arrivals can be reduced by 11.7% by re-distributing beds that are already available to the hospital.

3.1 Introduction

Overcrowding of hospital wards is a well known problem in the Danish health care sector. A report from the Ministry of Health [97] indicates that most regions of Denmark experience problems with overcrowding of hospital wards. In addition, the patient organization *Danish Patients* in corporation with *Danish Nurses Organization and the Danish Medical Association* reports that patient admission in hallways and depots is a recurrent necessity for a range of hospitals [4], and in which case both objective and subjective quality of care may suffer a great decrease [2, 119]. Hence, in order to provide patients with the

[1] Published in the European Journal of Operational Research [13]

best possible treatment, overcrowding should be reduced as much as possible.

An increasing number of Danish hospitals are developing methods to cope with overcrowding through capacity balancing, where patient relocation, is coordinated using daily capacity meetings, as well as dedicated staff for *patient flow* coordination [7]. Using such methods, some hospitals have succeeded in significantly decreasing the number of patients hospitalized to alternative locations. The hospitals relocate patients from wards with overcrowding to wards where sufficient nursing resources are still available, and thus match resources with demand. However, we conducted interviews with a specific hospital and found that this approach entails costs for both planning, relocation of patients and some decrease in quality of care. In this case, quality of care is decreased due to a mismatch between the optimal type of care and what type of care is alternatively offered to the patient. Hence, a problem arises containing *two* different types of penalty for the hospital management to consider. First, there is the tangible cost of spending man hours on defining and implementing a plan, and secondly, management have to consider the risk of inducing a lower quality of care, either through placing patients in buffer beds or relocating patients to other wards.

The objective of this study, is to provide hospital management with a tactical decision tool, capable of optimizing the match between resources and demand. We focus on a specific case where patients are always relocated whenever ward resources are depleted. The main methodological approach will be mathematical modeling. More specifically, we model bed occupancy using a homogeneous continuous-time Markov chain, and optimize the response using a local search heuristic.

In Section 3.2 we present the specific problem of this study. In Section 3.3 our solution approach is presented, divided into two parts. The first part describes the Markov chain, we use to model patient flow behavior. The second part connects this Markov chain model to a local search heuristic. Section 3.4 demonstrates the usability of our solution approach for a specific hospital case, and tests on a number of different parameter settings are presented. Lastly, we present our conclusion in Section 3.5.

3.1.1 Literature Review

Modeling and optimization of hospital bed utilization is a recurrent topic dating back to Newsholme, 1932 [94]. The specific problem structure differs from one study to another, however, all focus on one of three major objectives: (1) Testing scenarios [58, 11], (2) deriving the required number of beds for one or more wards [67, 103, 146, 59, 60, 62, 102], or (3) balancing beds with demand [41, 40, 83]. In achieving these, two methodological aspects are considered: (1) The methods used to *model* the system in focus, and (2) the methods used to *study* and *optimize* the system.

Different approaches of modeling the system are known from the literature. These are usually either simulation [58, 67, 146], queueing theoretic approaches [59, 60, 62, 102, 83, 40], or a mixture of these [11, 103, 41].

In Goldman et al., 1968[58], utilization and costs are tested for various bed allocation policies using a simulation model. Harris, 1984 [67], develops a simulation model to assist decision making in the area of operating theatre time tables and the resultant bed requirements. Lastly, Zeraati et al., 2005 [146], use a statistical simulation to estimate the number of required beds for an obstetrics ward.

In the area of queueing theoretic models, two studies by Gorunescu et al., 2002, and Li et al., 2009 [59, 60, 83], exploit $M/PH/c/N$ and $M/PH/c$ models, to assess a mixture of patient flow. Furthermore, Green, 2002 [62], use an $M/M/s$ model to estimate bed availability in different intensive care and obstetrics units, and Pendergast et al. 1988 [102], use clinical judgment and basic probability theory to derive future hospital bed requirements. Lastly, Cochran et al., 2008 [40], develops a queueing network model that is implemented as a capacity balancing tool between different hospital units.

Exploiting the use of both simulation and queueing theory, Cochran et al., 2006 [41], use queueing networks to assess the flow between units of an obstetrics hospital, and define utilization targets. A Discrete-event-simulation model is then used to maximize the flow. A related approach is used in Akkerman et al., 2009 [11], where Markov chain theory and simulation is used to evaluate a number of different management scenarios. This specific Markov chain model is found to produce useful insight into the theoretical number of required beds, but a simulation model is required to derive the amount of patient rejections.

The second methodological aspect that is considered in most studies, is studying and optimizing the system in focus. Naturally, scenario testing is more straightforward, whereas bed requirement or capacity balancing would suggest the application of a more elaborate approach. Here, we found only a few studies [103, 83] that exploit advantages of heuristic or mathematical programming elaborately, leaving this area rather unexploited. In Pinto et al., 2014 [103], a simulation-optimization model is developed to analyze dynamic features of the system and find the best configuration of beds. Moreover, Li et al., 2009 [83], applied a $M/PH/c$ model from Gorunescu et al., 2002 [59] in a multi-objective goal programming model to reallocate beds.

Two studies in the area of capacity balancing that are rather similar to this paper, are uncovered [41, 40]. However, in case of overcrowding, none of these studies modeled the effect of patients being relocated to alternative locations. In this paper, we present an approach to balance capacity in a system of queues, where patients are relocated when capacity is insufficient. To achieve this, we use a homogeneous continuous-time Markov chain model.

A range of studies, using Markov chains to model patient flow, have already

been conducted [19, 29, 48, 142, 147]. As relevant examples, Broyles et al., 2010 [29], predicts distribution and expected admissions, and Bartolomeo et al., 2008 [19], determines the probabilities of readmission for two different patient categories. However, none of these exploit the advantages of Markov chains, to model patient relocation, and subsequently use these models to optimize the system.

3.2 Problem Description

In this study, we consider a Danish hospital where an organizational structure for patient relocation has been fully implemented.

That is, even though minor changes in the distribution of resources might take place on a daily basis, most actions to avoid overcrowding are performed using patient relocation. Any greater changes in the distribution of resources are not practical if they occur too frequently, and are thus considered more as a tactical decision. Deciding on the best allocation of resources, is therefore an important decision, as the result will affect how patients are hospitalized, and the hereto related costs, through a period of several months.

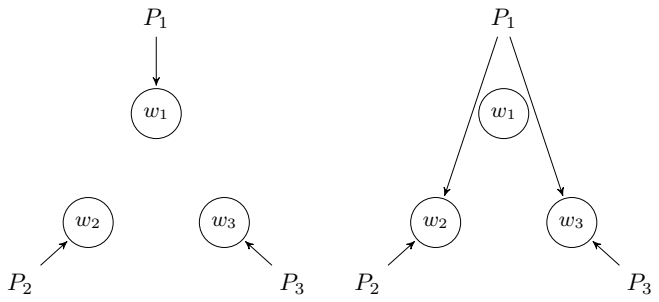
For this reason, the decision this study will focus upon, is how resources should be allocated among the hospital wards. As hospitalizations are usually dependent on a range of different resources, we assume that *one* hospitalization can only take place when *one* "sufficiently" staffed and equipped bed is available. That is, we disregard the possibility that a hospitalization may in some instances take place without sufficient staff or equipment. Thus, if an entrance to a ward is restricted by the lack of resource units, we assume that an alternative ward always exists somewhere else. We have found through interviews with hospital employees that this is a reasonable assumption.

Taking all of the above considerations into account, the overall goal of this study, is to develop a mathematical model that can be used to efficiently minimize the number of rejections at *preferred* wards, by changing the distribution of bed resources.

For the remaining of this paper, a patient hospitalization at a preferred ward, will be denoted as a *primary hospitalization*. A patient relocation to an alternative ward, will be denoted as a *secondary hospitalization*. In the same way, patient blocking at a preferred ward, is denoted a *primary rejection*, as well as patient blocking at the alternative ward is denoted a *secondary rejection*.

3.2.1 Dynamics of the System

As mentioned above, wards have limited resources, and as a result, arriving patients are relocated whenever resources have been depleted (no staffed and equipped beds are available). During such a relocation, patient characteristics



(a) All wards are available. (b) Ward 1 is blocked.

Figure 3.1: Graphical representation of the hospitalization procedure. Patient types are P_1, P_2 and P_3 for which the preferred hospitalization is at ward w_1, w_2 and w_3, respectively. (a) represents the system under regular load, and (b) the result if ward 1 was to be fully loaded.

are required to match with the specialization of the alternative ward. Thus, relocating a patient to an arbitrary free ward, is not always a feasible solution.

We interpret these hospitalization operations as a queueing system with N different patient types, arriving at N parallel service stations. The number of servers at each station is equal to the number of staffed and equipped beds at each ward. If all servers are occupied at a station the station is blocked, but a queue is not created. Instead, arrivals will be distributed with a probability to other stations, or disappear from the system entirely. This is illustrated in Figure 3.1, where a system of $N = 3$ patient types and wards is (a) under regular load, and (b), blocked for ward 1.

Due to these system operations, resources allocated to a ward will not only affect the amount of primary rejections, but also the amount of secondary hospitalizations at that ward. Moreover, notice that treatment time is tied to the patient type, and therefore independent of the ward in which hospitalization takes place. Wards will therefore experience a mixture of different patients with different lengths of stay.

3.3 Modeling & Solution Approach

To solve the problem of optimizing the distribution of beds, we model the ward occupancy density functions using a homogeneous continuous-time Markov chain. This model approach, is presented in Section 3.3.1. From the density functions, we derive the specific probabilities of wards blocking, followed by the overall expected number of arriving patients experiencing a primary rejection. This is used as our objective value, as the system is optimized using a local search heuristic. The specific heuristic we use, is presented in Section 3.3.2.

3.3.1 A Homogeneous Continuous-Time Markov Chain

As mentioned in Section 3.2, we consider N patient types, $i \in \{1, 2, \cdots, N\}$, as well as N ward types, $j \in \{1, 2, \cdots, N\}$. To model the ward occupancy density functions for each ward, we introduce a homogeneous continuous-time Markov chain (CTMC) model with state $s = (w_{11}, w_{21}, \cdots, w_{ij}, \cdots, w_{NN})$ and state space S, where w_{ij} is the number of type i patients hospitalized in ward j. Let M_j define the amount of allocated beds to ward j. Hence, M_j is the maximum amount of patients that may be hospitalized in ward j. Further, let f_j be the number of free beds at ward j, so $f_j = M_j - \sum_{i \in I} w_{ij}$. For the purpose of presenting our modeling approach, we include f_j in the state representation to get:

$$s \quad = \quad \left[\begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N1} & w_{N2} & \cdots & w_{NN} \end{pmatrix}, (f_1, f_2, \cdots, f_N) \right] \quad \in S$$

Note that f_j is otherwise redundant to the model. Now, λ_i is the arrival rate of patient type i, and μ_i the service rate of patient type i. We assume patients arrive according to a Poisson process and that inter-service time distributions are exponentially distributed. The reader should notice that if the latter does not hold, rejection systems, such as this, are in general robust to the distribution of inter-service times [28, p. 121]. In addition, we statistically test the CTMC model fit for a specific case-hospital in Section 3.4.1.

Let $p(f_1, f_2, \cdots, f_N)_{ij}$ define the fraction of patients of type i that are hospitalized in ward j as function of the number of free beds at all wards in the system, f_1, f_2, ... and f_N. Let Q define the transition rate matrix of the CTMC, with q_{ss^*} the transition rate from a current state $s \in S$ to a new state $s^* \in S$. In the following, $p(f_i = 0, f_j > 0, \cdots, f_N > 0)$ is abbreviated $p(f_i = 0)$, just as $p(f_i = 0, f_k = 0, f_j > 0, \cdots, f_N > 0)$ is abbreviated $p(f_i = 0, f_k = 0)$, and so on. Moreover, we refer to a new state $s^* = (\cdots, w_{ij}+1, \cdots, f_j-1, \cdots)$ to indicate the arrival of a patient i to a ward j, and $s^* = (\cdots, w_{ij}-1, \cdots, f_j+1, \cdots)$ for a corresponding discharge. The transition rates are then,

$$q_{ss^*} = \begin{cases} \lambda_i & \text{if } s^* = (\cdots, w_{ii}+1, \cdots, f_i-1, \cdots) \text{ and } f_i > 0 \quad \forall i \in I \\ \lambda_i p(f_i = 0)_{ij} & \text{if } s^* = (\cdots, w_{ij}+1, \cdots, f_j-1, \cdots) \text{ and } f_i = 0, f_j > 0 \quad \forall i, j \in I, i \neq j \\ \lambda_i p(f_i = 0, f_k = 0)_{ij} & \text{if } s^* = (\cdots, w_{ij}+1, \cdots, f_j-1, \cdots) \text{ and } f_i = 0, f_k = 0, f_j > 0 \quad \forall i, j, k \in I, i \neq j \neq k \\ \vdots & \vdots \\ \lambda_i p(f_i = 0, f_k = 0, \cdots, f_l = 0)_{ij} & \text{if } s^* = (\cdots, w_{ij}+1, \cdots, f_j-1, \cdots) \text{ and } f_i = 0, f_k = 0, \cdots, f_l = 0, f_j > 0 \\ & \forall i, j, k \cdots l \in I, i \neq j \neq k \neq \cdots \neq l \\ \mu_i w_{ij} & \text{if } s^* = (\cdots, w_{ij}-1, \cdots, f_j+1, \cdots) \text{ and } w_{ij} > 0 \quad \forall i, j \in I \end{cases}$$

where all other transition rates, q_{ss^*}, are 0.

Notice, as treatment times differ between patient types, the state definition contains an element for every combination of patient type and ward. The variables w_{ii} count primary hospitalizations, whereas the variables w_{ij} count

secondary hospitalizations. The model can only jump to a state, where the number of secondary hospitalizations is increased, if capacity is full at the preferred ward. For instance if $N = 3$ and $M_1 = 10$, $M_2 = 15$ and $M_3 = 20$, $s = (w_{11}, w_{21}, w_{31}, w_{12}, w_{22}, w_{32}, w_{13}, w_{23}, w_{33}) = (7, 2, 1, 1, 4, 2, 3, 2, 10) \rightarrow s^* = (7, 2, 1, \mathbf{2}, 4, 2, 3, 2, 10)$ is allowed, because ward 1 is full. However, $s = (7, 2, 1, 1, 4, 2, 3, 2, 10) \rightarrow s^* = (7, 2, 1, 1, 4, 2, 3, \mathbf{3}, 10)$ is not possible, as ward 2 is still open.

The transition rate depends on how many other wards are blocked. $s = (7, 2, 1, 1, 4, 2, 3, 2, 10) \rightarrow s^* = (7, 2, 1, \mathbf{2}, 4, 2, 3, 2, 10)$ has rate $q_{ss^*} = \lambda_1 p(f_1 = 0)_{12}$, as only ward 1 is blocked. Now, $s = (7, 2, 1, 1, 4, 2, 3, 2, 15) \rightarrow s^* = (7, 2, 1, \mathbf{2}, 4, 2, 3, 2, 15)$ has rate $q_{ss^*} = \lambda_1 p(f_1 = 0, f_3 = 0)_{12}$, as both ward 1 and ward 3 are blocked.

The total state space size, $|S|$, of the CTMC is the product of N polynomials of the order N, as shown in (3.1).

$$|S| = \prod_{j=1}^{N} \left(\frac{1}{N!} \cdot \prod_{i=1}^{N} (M_j + i) \right) \tag{3.1}$$

Let us consider a case where $N = 3$, and $M_1 = 27$, $M_2 = 23$ and $M_3 = 24$. Then, from (3.1), the state space, S, has a size of $|S| = 30,876,300,000$ states – which is rather difficult to cope with computationally. Thus, in order for our CTMC to be applicable for even small cases, a rather large fraction of the state space needs to be truncated. To attain this, we use two recursive procedures presented in the following Section 3.3.1.

The Truncation Procedures

Let u_{ij} be an upper bound on the number of patients of type i that is hospitalized in ward j (w_{ij}) for $i \neq j$, so $\sum_{k=u_{ij}+1}^{M_j} Prob\{w_{ij} = k\}$ is *sufficiently* small, where $Prob\{w_{ij} = k\}$ is the probability of attaining a state where $w_{ij} = k$. Taking this idea further, we also let L_j and U_j define the lower and upper bounds on the total amount of patients hospitalized in ward j. In this case, L_j and U_j are chosen so $\sum_{k=L_j}^{U_j} Prob\{\sum_i w_{ij} = k\}$ is *sufficiently* large, but the number of truncated states are maximized. Here, $Prob\{\sum_i w_{ij} = k\}$ is the probability of attaining a state where the sum of patients in ward j is equal to k. Let ϕ_j be the number of free slots at ward j in the truncated system, then $\phi_j = U_j - \sum_i w_{ij}$. Thus as $U_j \leq M_j$, we have that $\phi_j \leq f_j$.

Our goal is then to adjust u_{ij}, L_j and U_j, so reasonable accuracy is maintained, within the practical limits of computing the probability distribution. To attain this, we notice that the hospitalization of patients at each ward is closely related to an $M/M/c/c$ queueing system, cf. Figure 3.2. That is, a queueing system with capacity equal to the number of beds. The probability that there are n beds occupied in such a system can be derived using (3.2),

$$p_n = \frac{(\lambda/\mu)^n / n!}{\sum_{i=0}^{c} (\lambda/\mu)^i / i!} \tag{3.2}$$

where λ is the arrival rate, μ the service rate, and c the number of beds in the system [28, p. 121]. We use (3.2) to determine bounds on the total amount of occupied beds at each ward, L_j and U_j, as well as for each secondary hospitalization pair, u_{ij}. For the latter, consider that w_{ij} is stochastically smaller or equal to the occupancy in an $M/M/c/c$ system where the arrival rate is the maximum fraction of arriving type i patients to ward j, $\lambda_i \cdot \max\{p(\cdot)_{ij}\}$, and service rate μ_i. The probability mass of such a system, derived using (3.2), will be at least as shifted in positive direction as the marginal probability mass of w_{ij} in the CTMC. We refer to this $M/M/c/c$ system as the *right-shifted* distribution. Letting τ ($0 \leq \tau \leq 1$) define a truncation tolerance, the upper bound, u_{ij}, is derived using Algorithm 5.

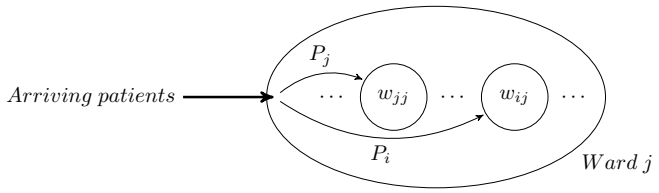


Figure 3.2: Graphical representation of patients hospitalized in ward j. Patients of different types are hospitalized as long as $\sum_i w_{ij} < M_j$, where M_j corresponds to the parameter c in (3.2).

Algorithm 5 Procedure for deriving u_{ij} for $i \neq j$

$lamdba \leftarrow \lambda_i \cdot \max\{p(\cdot)_{ij}\}$ ▷ Initialize
$mu \leftarrow \mu_i$
$c \leftarrow M_j$
$right \leftarrow erlangB(c, lambda, mu)$ ▷ Calculate distribution using (3.2) and save as an array
$st \leftarrow length(right)$
while $sum(right) \geq 1 - \tau$ **do**
    $right[st] \leftarrow 0$
    $st \leftarrow st - 1$
end while
$u \leftarrow st + 1$ ▷ The final bound is the number of non-zero elements plus 1
return u

Here, $right$ represents the right-shifted distribution for w_{ij}. The upper bound, u_{ij}, is then determined by recursively truncating $right$ starting at the highest occupancy, and lastly adding one to ensure that the least probability mass larger or equal to $1 - \tau$ is left.

In deriving L_j and U_j we consider both a lower and an upper bound of occupancy in ward j. Therefore, in order to ensure sufficient probability mass in the CTMC, and derive the maximum number of states that may be truncated

from ward j, we have to consider both a left-shifted *and* right-shifted distribution at the same time. The procedure we use to derive L_j and U_j, is presented below:

1. Determine the minimum service rates in ward j: $\mu_{MIN} \leftarrow \min_{i \in I}\{\mu_i\}$.
2. Determine the maximum arrival rate to ward j: $\lambda_{MAX} \leftarrow \lambda_j + \sum_{i \in I \setminus \{j\}} \lambda_i \cdot \max\{p(\cdot)_{ij}\}$.
3. Calculate left- and right-shifted distributions using (3.2).
   (a) $right \leftarrow erlangB(M_j, \lambda_{MAX}, \mu_{MIN})$.
   (b) $left \leftarrow erlangB(M_j, \lambda_j, \mu_j)$.
4. Truncate states constrained by τ: $L_j, U_j \leftarrow ipmodel(right, left, \tau)$.

Notice that our procedure lastly takes an Integer Programming model, $ipmodel()$, to maximize the number of truncated states. We propose to formulate this as a *Knapsack Problem* variation, minimizing the number of states in the truncated system constrained by the probability mass tolerance τ.

This concludes the approach we use to derive u_{ij}, L_j and U_j. Notice, how the resulting transition rate matrix, Q, will be dependent on whether $U_j < M_j$ or $U_j = M_j$, leading to different representations of the matrix.

Computing the State Probability Distribution

We have derived a method for reducing the state space as a function of the tolerance τ, and are therefore set to generate the transition rate matrix Q. We assume that most non-acute wards will have long expected length of stay relative to the respective fluctuations in arrival rate. We further assume that most arrivals and discharges occur during the day, causing the system to be "inactive" during the night, so any remaining time-dependency will be negligible in the scope of deriving a long-term allocation of beds for the hospital. Thus, we consider the CTMC as a steady-state process. Now let π define the steady-state probability distribution of the CTMC. Then, we are faced with solving the global balance equations in (3.3),

$$\pi Q = 0 \tag{3.3}$$

where $\|\pi\|_1 = 1$. We have found that a solution to (3.3) can be derived within reasonable runtime using the method of *successive overrelaxation* [123, p. 311]. That is, (3.3) is written on the form $Ax = b$ by transposing, so we get: $Q^T\pi^T = 0$. Further, $Q^T\pi^T = (D - X - Y)\pi^T = 0$, where D, X and Y are the diagonal, lower- and upper- strictly triangular matrices of Q^T. Let x^k be the $k'th$ iteration solution to π^T. Then we can recursively derive π^T, using (3.4),

$$x^{k+1} = (1 - \omega)x^k + \omega\{D^{-1}(Xx^{k+1} + Yx^k)\} \tag{3.4}$$

until convergence. The relaxation parameter, ω, may be adjusted to ensure the fastest rate of convergence. As our case is dependent on different representations of Q, and we want our implementation to be flexible, we chose to conduct a range of tests to search for a fixed relaxation parameter that would result in a reasonable convergence time for "most" cases. We calculated the *iteration matrix* $H_w = (D - \omega X)^{-1}[(1 - \omega)D + \omega Y]$. We then adjusted ω to maximize the distance between the unit dominant and subdominant eigenvalue of H_w, with a view to maximize the convergence rate. It was found that a high distance could be obtained with a relatively high relaxation parameter – usually around 1.7 to 1.8. Thus for the remaining of this paper, $\omega = 1.75$.

Regarding the question of *when* convergence has occurred, we decided to check this on the largest relative tolerance $\delta = max_i(\left|x_i^k - x_i^{k-m}\right| / \left|x_i^k\right|)$. Where m is set to increase as δ decreases – recall $\lim_{\delta \to 0} x^k = \pi^T$, and thus the rate of convergence is expected to decrease as x^k is closing in on π^T.

To assess our approach, we conducted a series of tests for $N = 3$, $M_1 = 27$, $M_2 = 23$ and $M_3 = 24$, and different settings of the truncation parameter τ. In Table 3.1, the total runtime of our approach implemented in `Java`, along with state space sizes, are presented for $\tau = 0.05, 0.01$ and 0.001.

Each of these settings were assessed by comparing the respective marginal distributions of π – that is, the distribution of how many beds are expected to be occupied for each ward. Obviously, the tails approach zero as the truncation is relaxed. However, the algorithm only takes 69 seconds to finish for $\tau = 0.05$, against 1,947 seconds for $\tau = 0.001$. Additionally, in case we are only interested in the blocking probabilities, we would be able to make do with the largest truncation value – given that we always end up with a CTMC model representation where $U_j = M_j \quad \forall j \in I$. However, to gain a more generic use of our model, we find it more appropriate to use $\tau = 0.01$.

τ	Total Runtime (s)	$\lvert S \rvert$
0.05	69	517,000
0.01	483	1,358,760
0.001	1,947	3,563,520

Table 3.1: Results from adjustment of τ.

3.3.2 A Heuristic Optimization Model

In Section 3.3.1 we presented an approach to model the ward occupancy for N wards and correspondingly N patient types. We now consider the number of beds, available to ward i, M_i, as a decision variable that may be adjusted to optimize the overall system performance. In general, we consider the following optimization problem:

$$\begin{aligned} \text{min.} \quad & f(\boldsymbol{M}) && \text{(3.5a)} \\ \text{s.t.} \quad & \\ & \textstyle\sum_{i \in I} M_i = \Theta && \text{(3.5b)} \\ & M_i \geq 1 \; \forall i \in I && \text{(3.5c)} \\ & M_i \in \mathbb{N} \end{aligned}$$

Where, as previously defined, I is the set of wards. Here, (3.5b) ensures that all available resources, Θ, are utilized. Moreover, (3.5c) ensures that wards contain at least one bed. The objective function (3.5a) evaluates the system performance as a function of $\boldsymbol{M} = (M_1 \; M_2 \; \cdots \; M_N)^T$, where in this case, a *large* value indicates a *poor* performance. As shown in the following, the objective function can easily be replaced and customized to the specific hospital preferences. In this study, we propose an objective value that increases as more "work" is spent on relocating patients. Consider $f(\boldsymbol{M}) = \sum_{i \in I} \pi_i^B(\boldsymbol{M})$, where $\pi_i^B(\boldsymbol{M})$ is the probability of all beds being occupied in ward i, with beds distributed as in $\boldsymbol{M}$. In this case, we would get some kind of measure for the total amount of work – recall when $\pi_i^B(\boldsymbol{M})$ increases, so does the amount of relocated patients from ward i. However, the expression does not incorporate the weight of patient types arriving with different rates. So we insert λ_i, to get (3.6), the total expected number of primary rejections.

$$f(\boldsymbol{M}) = \sum_{i \in I} \lambda_i \pi_i^B(\boldsymbol{M}) \tag{3.6}$$

Returning to the idea that (3.2) can be used to approximate the occupancy at a single ward, we have a way to estimate $f(\boldsymbol{M})$. Specifically, we use (3.2) to estimate $\pi_i^B(\boldsymbol{M})$, by calculating the blocking probability $p_c = B(c, \lambda/\mu)$ – known as the *Erlang-B* formula.

Inserting $B(c, \lambda/\mu)$ into (3.6), we are now able to derive an estimate of the objective value using (3.7). Doing so, gives us the opportunity to derive an estimate of the optimal solution in a few seconds.

$$\hat{f}(\boldsymbol{M}) = \sum_{i \in I} \lambda_i B(M_i, \lambda_i/\mu_i) \tag{3.7}$$

Now, from (3.5b) we have that $\sum_{i \in I} M_i = \Theta$. Therefore, $M_N = \Theta - \sum_{i \in I \setminus \{N\}} M_i$, reducing (3.7) to a function of $N-1$ variables, $\hat{f}(M_1, M_2, \cdots, M_{N-1})$. Let $\hat{f}_{M_i}(\cdot)$ be the i'th partial derivative of $\hat{f}(M_1, M_2, \cdots, M_{N-1})$, with the derivative of $B(M_i, \lambda_i/\mu_i)$ presented in (A.1) in Appendix A.1. The horizontal tangent plane of $\hat{f}(M_1, M_2, \cdots, M_{N-1})$ can then be found from the system of equations: $\hat{f}_{M_1}(\cdot) = 0 \wedge \hat{f}_{M_2}(\cdot) = 0 \wedge \cdots \wedge \hat{f}_{M_{N-1}}(\cdot) = 0$. We solve this, using the *Newton-Raphson method*.

Now, recall that the difference between (3.6) and (3.7), is the relocation of patients from fully occupied to free wards. Therefore, as the probabilities $p(\cdot)_{ij}$ from the CTMC model decrease, (3.6) approaches (3.7). In other words, an

optimal solution derived using (3.7) is likely close to the optimal solution using (3.6). To locate the optimal solution to the optimization problem (3.5a)-(3.5c), an idea would therefore be to *slowly* change the solution configuration, starting with an initial guess derived from the estimate (3.7).

Let $N(\boldsymbol{M})$ define the "neighborhood" of the bed distribution $\boldsymbol{M}$, and still consider that $M_N = \Theta - \sum_{i \in I \setminus \{N\}} M_i$, so now $\boldsymbol{M} = (M_1\ M_2\ \cdots\ M_{N-1})^T$. Then, $(\boldsymbol{M} + \boldsymbol{\nu}) \in N(\boldsymbol{M})$, where $\|\boldsymbol{\nu}\| = 1$ and the elements $\nu_i \in \{0, -1, 1\}$. This leads to a maximum neighborhood size of $|N(\boldsymbol{M})| = N^2 - 1$ or $O(N^2)$. Hence, in case $N = 3$, $|N(\boldsymbol{M})| = 3^2 - 1 = 8$ solutions.

Consider if $\boldsymbol{M}^*$ is the currently best known solution to (3.5a)-(3.5c), then an idea would be to systematically check $N(\boldsymbol{M}^*)$ for an even better solution, and update $\boldsymbol{M}^*$ in case such a solution is found. This leads to the local search heuristic presented in Algorithm 6.

Algorithm 6 Heuristic to optimize the bed requirements problem in (3.5).

$M^* \leftarrow init(M^0)$ ▷ Initialize using the horizontal tangent plane of (3.7)
$f^* \leftarrow objval(M^*)$
$N \leftarrow generateneigh(M^*)$ ▷ Generate list of neighborhood in random order
$C \leftarrow \emptyset$
$j \leftarrow 0$
while $j < length(N)$ and $elapsedtime < timelimit$ **do**
    $j \leftarrow j + 1$
    $M \leftarrow N[j]$
    if $checkbanned(C, M) == false$ **then** ▷ Check banned or constraint violation.
        $C \leftarrow add(C, M)$ ▷ Add M to the list of banned solutions.
        $f \leftarrow objval(M)$
        if $f < f^*$ **then**
            $f^* \leftarrow f$ ▷ Update values
            $M^* \leftarrow M$
            $N \leftarrow generateneigh(M^*)$ ▷ Generate the new neighborhood, again in random order
            $j \leftarrow 0$
        end if
    end if
end while

The heuristic progresses by firstly generating an initial solution from the horizontal tangent plane of (3.7). This is conducted using the function $init()$ that, based on the Newton-Raphson method, takes an initial guess M^0. The "raw" output is most likely not integral, so we round to the integer solution yielding the lowest objective value. Next, the currently best known objective value, f^*, is calculated.

Then, $generateneigh()$ is used to generate a list of the entire neighborhood. For larger cases, a probabilistic candidate list might be more appropriate, choosing a random fraction of the solutions in $N(\boldsymbol{M})$. Elements of the list should in any case be placed in random order.

Due to the local progression of the heuristic, and a relatively long function evaluation time, we further introduce a list of *banned* solutions, C. As the heuristic can only move one step at a time, there will always be an overlap between the neighborhood of iteration k and $k+1$. For this reason, we add all previously evaluated solutions to a list (line 10, Algorithm 6), to ensure that we do not spend time on evaluating a solution more than once.

3.4 Implementation & Results

In this section, we directly implement the methods from Section 3.3 to obtain an improved distribution of beds for a case-hospital. In modeling the system behavior, we have limited our scope to the hospitalization of patients to the medical area of the hospital. More specifically, we focus on patient flow in gastrology, pneumology, endocrinology and geriatrics, respectively. For the case hospital, these areas make up *three* different wards and correspondingly *three* different patient types.

In Section 3.4.1, we present the data obtained from the case-hospital and statistically test our homogeneous continuous-time Markov chain (CTMC) model. Next, Section 3.4.2 presents the implementation of our solution approach. Lastly, we assess the robustness of Algorithm 6, and investigate the solution behavior when the CTMC model parameters are adjusted. This is presented in Section 3.4.3.

3.4.1 Case & Data Description

The patient flow was investigated through interviews with hospital staff. Furthermore, we retrieved data from the period of 01-05-2014 to 30-04-2015 on patient arrival and discharge times. From this, we were able to categorize patients on diagnosis and thus also treatment type, giving us the opportunity to determine preferred and alternative wards.

Arrival Rates

Patient hospitalization data was used to derive hourly arrival rates for each of the three patient types, showing clear repetitive patterns on a weekly scale of the hourly arrival rate. In Figure 3.3, the empirical hourly arrival rates are presented for all patient types. As expected for non-acute wards, most patients are hospitalized during the daytime, whereas an almost negligible fraction of patients arrive during the night. Further, the arrival rates seem to slightly decrease during the weekend, and regain its level starting Monday. The empirical

average arrival rates were estimated to $\lambda_1 = 5.42$, $\lambda_2 = 3.96$, $\lambda_3 = 2.52$ patients per day, respectively.

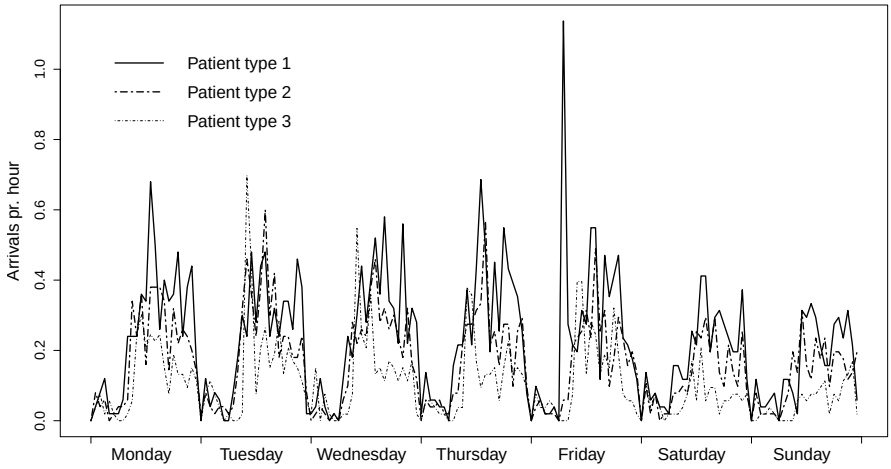


Figure 3.3: Empirical hourly arrival rate. Presented for each of the three patient types.

Service Rates

To derive service rates for each of the three patient types, we calculated the *Length of Stay* (LOS) using the same patient hospitalization data, as was used to derive the arrival rates. Time-dependency was checked on a daily level by deriving the average hourly LOS from time of arrival. Performing a graphical representation showed no signs of seasonality, neither did an estimate of the autocorrelation function. Regarding load-dependency, we did not obtain sufficient data to confirm nor reject such behavior. For the case-hospital, capacity meetings are often held to ensure that patients are immediately relocated upon blocking. As a result, neither LOS increase nor early-discharge due to overcrowding is rarely the case. The overall distribution of LOS was investigated graphically, where the patient type 1 and 2 distributions show close similarity to an exponential distribution (See Figure 3.4). With a longer average LOS, the patient type 3 distribution has probability mass that is moved more to the right, quite similar to a gamma distribution.

Due to the similarities that was found between patient type 1 and 2, we tested their difference in mean LOS using a Wilcoxon rank-sum test [139]. With a p-value of 0.2105, we found no significant difference in mean LOS between the two patient types. Figure 3.4 suggests that there is no difference in statistical distribution either. For patient type $i \in \{1, 2, 3\}$, the resulting

service rate ($1/LOS_i = \mu_i$) was estimated based on an empirical average to $\mu_1 = \mu_2 = 0.19$ and $\mu_3 = 0.11$ patients per day, respectively.

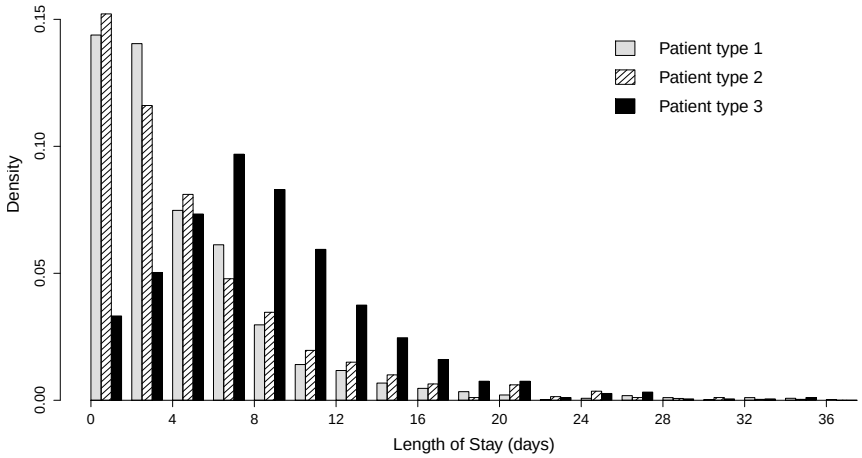


Figure 3.4: Empirical distribution of *length of stay*, measured in number of days. Presented for each of the three patient types.

Relocations

We investigated the secondary hospitalization options for each of the three patient types. From data, we obtained the 80% most common diagnoses for each patient type, and for each of these diagnoses, hospital staff identified the alternative locations they would usually offer to these patients. This allowed us to draw a picture of how relocated patient are usually distributed. Here, we found that patients have secondary hospitalization options both within and outside the three wards; hence, in case of blocking, a fraction of patient will always be lost from the system. Moreover, we found that patients would usually have a third hospitalization options – however, for this case, we found it reasonable to assume that a third hospitalization options is always situated outside the system. The resulting relocation probabilities are presented in Table 3.2, showing that a reasonably large fraction of patients has to be relocated elsewhere.

Other Characteristics of the System

From the distinct fluctuating arrival rate, one would naturally expect the level of hospitalized patients to be fluctuating as well. Figure 3.5 shows the empirical probability of a patient being discharge as function of hour of the week. As expected for non-acute patients, discharges mainly occur on weekdays during

P_i/w_j	1	2	3	Other
1	-	0.05	0.23	0.72
2	0.10	-	0.27	0.63
3	0.06	0.00	-	0.94

Table 3.2: Probability that patient type $i \in \{1, 2, 3\}$ (P_i), is relocated to ward $j \in \{1, 2, 3\}$ (w_i), in case ward i is blocked.

the daytime, with a negligible number of patients discharged during the night. Comparing with the arrival rates in Figure 3.4, we notice that the system is mainly "active" between 07:00 and 23:00.

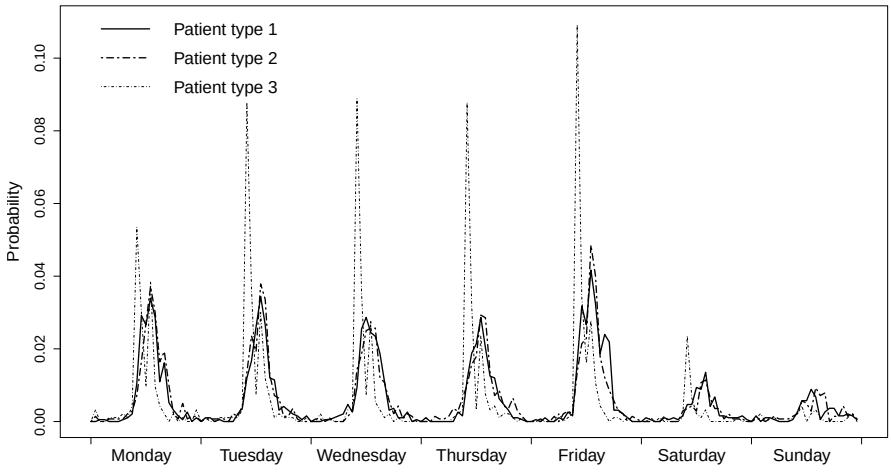


Figure 3.5: Empirical probability of a patient being discharged as function of hour of the week. Presented for each of the three wards.

Observations from 14-09-2015 to 31-10-2015 were obtained to investigate the time-dependent behavior of ward occupancy in the system. Figure 3.6 shows the average number of occupied beds every 8'th hour during the week. From here, we notice some time-dependent behavior as the occupancy is usually lower during the middle of the day. This behavior repeats on a daily basis, with a small overall decrease during the weekend for ward 2 and 3. Taking the time-dependency of hourly arrival rate and discharge hour into account might be necessary for purposes of accurately predicting the occupancy for each specific hour of the week. However, as our aim is to derive a long-term allocation of beds for the hospital, we consider the observed fluctuations as negligible for this case.

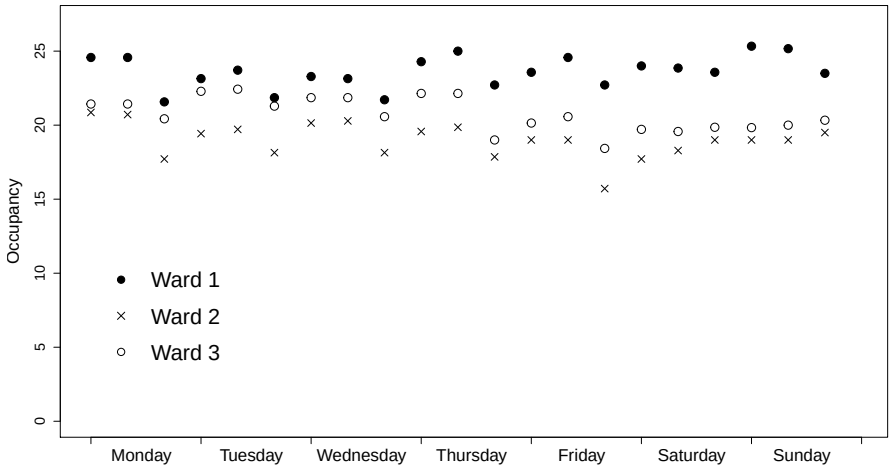


Figure 3.6: Empirical average number of occupied beds for each of the three wards. Observations were obtained for every 8'th hour of the week.

Truncation of the CTMC

For practical reasons we are often required to truncate the CTMC prior to implementation. We start this process by firstly considering the data obtained from the case-hospital. In Section 3.4.1 we found that $\mu_1 = \mu_2$; hence the number of patients of type 1 and 2, can be contained in only two "bins" of the state space. In other words, w_{11} and w_{12} are merged into w_{121}, as well as w_{21} and w_{22} are merged into w_{122}. Moreover, from Table 3.2, we have that $p(f_1, f_2, f_3)_{32} = 0$ in all cases, so w_{32} can be neglected. This results in the state representation:

$$s \quad = \quad \left[\begin{pmatrix} w_{121} & - & w_{13} \\ - & w_{122} & w_{23} \\ w_{31} & - & w_{33} \end{pmatrix}, (f_1, f_2, f_3) \right] \quad \in S$$

Now we apply the truncation procedures described in Section 3.3.1. We use the truncation parameter $\tau = 0.01$, as proposed in Section 3.3.1. For the case-hospital, the total number of beds $\Theta = 74$, so we calculate u_{ij}, L_j and U_j for any feasible value of $M_j \in \{0, 1, \cdots, \Theta - (N-1)\} \quad \forall j \in J = \{1, 2, 3\}$.

To illustrate the resulting truncation, the total state space size, $|S|$, for a non-truncated model with $N = 3$ wards, where $M_1 = 27$, $M_2 = 23$ and $M_3 = 24$ has $|S| = 30,876,300,000$ states. The truncated model, with the same settings, has $|S| = (\sum_i^{u_{31}}(U_1 - i - \max\{L_1 - 1; 0\} + 1))(U_2 - L_2 + 1)(\sum_{j=0}^{u_{13}} \sum_{k=0}^{\min\{U_3 - i; u_{23}\}}(U_3 - j - k - \max\{L_3 - i - j; 0\} + 1)) = 1,358,760$ states — a substantial reduction of the state space.

Statistical Testing of the CTMC

We conducted a statistical test to assess the CTMC model fitness with observations on ward occupancy. To our knowledge, there exists no standard technique to test the fitness of a CTMC with a complexity as considered in this study. Thus in this section we present a heuristic approach that combines a simulation of the CTMC behavior and compare this to hospital data on ward occupancy.

To begin with, our null-hypothesis is that the observed values are generated by the CTMC process. If that is the case, we would expect the observed frequency of occupied beds to be quite similar to the marginal distributions of π for each ward. A standard approach would be to test the observed frequencies against the corresponding expected frequencies from the CTMC using a *chi-squared* test. However, such as test would require each of the observed values to be independent, which is not the case here.

Let ω_j be the expected number of occupied beds from the CTMC for ward $j \in J$, where $J = \{1, 2, 3\}$. Then $\omega_j = \sum_{k=0}^{M_j} k \cdot \pi_{kj}$, where π_{kj} is the probability that ward $j \in J$ has k occupied beds. Further, let o_{ij} and e_{ij} be the observed and expected frequency of i occupied beds in ward j. Then, we define our test statistic as (3.8),

$$T = \sum_{j \in J} \sum_{i \in I} (o_{ij} - e_{ij})^2 / \omega_j \tag{3.8}$$

where $I = \{0, 1, 2, \cdots, M_j\}$ is the set of beds that can be occupied, and $J = \{1, 2, 3\}$ the set of wards. In order to quantify the fit of our CTMC model, we require a measure of how (3.8) relates to the model noise. For this reason, we introduce the simulated model residual (3.9), where y_{ij} is the simulated frequency of i occupied beds in ward j. Thus, by replicating (3.9), we determine the distribution of noise that is expected by our CTMC model, and then compare our results from (3.8) hereto.

$$z = \sum_{j \in J} \sum_{i \in I} (y_{ij} - e_{ij})^2 / \omega_j \tag{3.9}$$

We implemented the simulation of the CTMC model as a Discrete-Event-Simulation using `Matlab`. Replications of (3.9) were conducted $n = 30,000$ times (Appendices, Figure A.1).

A total of 432 (144 pr. ward) observations were obtained from the period of 14-09-2015 to 31-10-2015. Using these, we calculated $T = 0.45$. A fraction of 32.03% simulated residuals, scoring higher than T, were found. Thus, with a significance level of $\alpha = 0.05$, we *accept* the null-hypothesis.

The power of our test was evaluated by conducting a range of experiments where model parameters were adjusted until under 5% significance would be

obtained. Specifically, we adjusted the arrival rates proportionally with 10% increment at the same time.

Results are presented in Table 3.3, showing a 10% increase (Test 1) and 20% decrease (Test 3), were sufficient adjustments to gain less than 5% significance.

#	Change	λ_1	λ_2	λ_3	p-value
0	1.0	5.42	3.96	2.52	0.32
1	1.1	5.96	4.36	2.77	0.01
2	0.9	4.88	3.56	2.27	0.56
3	0.8	4.34	3.17	2.02	0.02

Table 3.3: Assessment of the power of our test. Conducted by proportionally changing the three arrival rate parameters (λ_i) and evaluating the resulting simulated p-value (p).

3.4.2 Optimizing the Case-Hospital

We now consider the heuristic in Algorithm 6 applied to the case-hospital. We initialize the heuristic, by handing $M^0 = (27\ 23)^T$ to $init()$. From the Newton-Raphson method, we get $M_1 = 31.80$ and $M_2 = 23.50$. The rounded integer solutions are then: (1) $f(\lfloor M_1^0 \rfloor, \lfloor M_2^0 \rfloor) = 1.473$, (2) $f(\lceil M_1^0 \rceil, \lceil M_2^0 \rceil) = 1.468$, (3) $f(\lfloor M_1^0 \rfloor, \lceil M_2^0 \rceil) = 1.470$ and (4) $f(\lceil M_1^0 \rceil, \lfloor M_2^0 \rfloor) = 1.467$. (4) returns the lowest value, so we set $M^* = (32\ 23)^T$, and proceed. The initial objective value is then $f^* = 1.603$ patients per day, and initial neighborhood $N = \{(33\ 23), (31\ 23), (32\ 24), (32\ 22), (33\ 24), (31\ 22), (31\ 24), (33\ 22)\}$. As the intention is to use our method as a *tactical* decision tool, a maximum time-limit of 4 hours is considered reasonable. The entire heuristic and its components are implemented in `Java`.

Each iteration is presented below, showing all function evaluations and how the list of banned solutions is updated progressively:

- *Iteration 1* – Checking: $f(31, 22) = 1.641$ and $f(33, 23) = 1.600$. $f(33, 23) = 1.600 < f^*(32, 23) = 1.603$, so we update, $f^* \leftarrow f$, to $f^* = 1.600$. The list of banned solutions is now: $C = \{(32\ 23), (31\ 22)\}$.

- *Iteration 2* – Checking: $f(34, 24) = 1.606$, $f(32, 22) = 1.623$, $f(34, 22) = 1.624$, $f(33, 22) = 1.620$ and $f(32, 24) = 1.592$. $f(32, 24) = 1.592 < f^*(33, 23) = 1.600$, so we update, $f^* \leftarrow f$, to $f^* = 1.592$. The list of banned solutions is now:
  $C = \{(32\ 23), (31\ 22), (34\ 24), (32\ 22), (34\ 22), (33\ 22), (32\ 24)\}$.

- *Iteration 3* – Checking: $f(31, 25) = 1.607$, $f(33, 24) = 1.596$, $f(31, 23) = 1.617$, $f(33, 25) = 1.603$, $f(32, 25) = 1.600$ and $f(31, 24) = 1.606$. After checking all available solutions in the neighborhood, f^* was not improved, so we **stop** and conclude that $M_1 = 32$, $M_2 = 24$ and $M_3 = \Theta - (M_1 + M_2) = 18$ is at least a local optimum. The total runtime was **1791.12** seconds (≈ 30 min.).

Notice, the final solution, $\boldsymbol{M}^* = (32\ 24)^T$, is actually an element in the initial neighborhood, and could have been selected by chance during the first iteration. Moreover, in generating the initial solution, $(32\ 24)^T$ would have been obtained by simply rounding to nearest integer.

Now, consider that the search space for three wards has a total size of $n = \sum_{i=1}^{\Theta-2}(\Theta - 1 - i) = (1/2)\Theta^2 - (3/2)\Theta + 1$. Thus for $\Theta = 74$, $n = 2628$ solutions. As n is reasonably low for this case, a complete enumeration of the search space is possible. To investigate how results from the heuristic relates to the optimal solution, we conducted a complete enumeration with the result presented in Appendices, Figure A.2. Interestingly, the objective function contains only a single extrema – a global minimum in $M_1 = 32$ and $M_2 = 24$. Hence, we can conclude that the solution found from the heuristic, $\boldsymbol{M}^* = (32\ 24)^T$, is in fact the *global* optimal solution to the problem. We ask the reader to notice that the procedure of complete enumeration spend 464,212.02 seconds (≈ 5 days and 9 hours) to complete. Thus, even though the procedure is possible, it is certainly not practical. The heuristic in Algorithm 6, solved the problem in just under 30 min. That is, 99.6% faster.

The optimal solution is compared to the current distribution of beds in Table 3.4. As for the current distribution, $f(\boldsymbol{M}) = 1.804$ patients per day, the optimal solution yields a 11.77% reduction in number of primary rejections. We notice for the current case, the highest probability of ward blocking, π_i^B, takes place in ward 1. Unfortunately, we find that patient type 1 has the highest arrival rate of 5.42 patients per day as well. Thus, it would be expected, in order to minimize $f(\boldsymbol{M})$, additional resources has to be pushed to ward 1 with a view to decrease π_1^B. Conversely, patient type 3 has the lowest arrival rate of 2.52 patients per day, and with $\pi_3^B = 0.161$, ward 3 is expected to reject 0.41 patients per day – 0.56 patients fewer than for ward 1. Turning to the optimal solution, we find the probability of rejection has been vastly increased for ward 3, but decreased for both ward 1 and 2. Further, we find that $\max_{i\in I}\{\lambda_i\pi_i^B\}$ has been decreased from 0.969 to 0.803 patients per day, and $\min_{i\in I}\{\lambda_i\pi_i^B\}$ decreased from 0.405 to 0.335 patients per day.

	Current			**Optimal**		
Ward	M_i	π_i^B	$\lambda_i\pi_i^B$	M_i	π_i^B	$\lambda_i\pi_i^B$
1	27	0.178	0.969	32	0.083	0.454
2	23	0.109	0.430	24	0.084	0.335
3	24	0.161	0.405	18	0.318	0.803
$f(\boldsymbol{M})$	-	-	1.804	-	-	1.592

Table 3.4: The optimal solution compared to the current distribution of beds. Presented with objective values, $f(\boldsymbol{M})$, beds M_i and blocking probabilities π_i^B. The product $\lambda_i\pi_i^B$ shows the expected daily amount of primary rejections for each ward.

3.4.3 Case Testing

With a view to investigate the solution behavior of our heuristic, we conducted a series of tests with various parameter adjustments. The hospital is planning to introduce a number of organizational changes, with the result of increased patient arrival rate, but additional overall bed capacity. Thus in our last test, we demonstrate how our approach may be used as a tool to assess future changes to the organization.

We conducted a total of five different basic tests, where patient flow or available resources were changed. In Table 3.5 the parameters that were subject to change are presented in bold font, the rest are from the hospital case.

The results for each of Test 1-5 are presented in Table 3.6, with firstly the initial solution, then the optimal solution, and lastly data on the heuristic progression. The total number of function evaluations that are avoided as a result of the list of banned solutions are presented in the second last column.

Giving the five tests a closer look, we expect for Test 1-3 that an increase in arrival rate results in a corresponding increase in allocated resources. This behavior is found for each of the three tests, where resources are allocated to respond to the increased demand for primary hospitalizations.

In Test 4 nothing was changed but the total amount of available beds Θ. We conducted this test, to assess the potential improvements caused by a relatively small increase in resources. We find that, as more resources are available in the system, additional surplus is created and the fractional distribution of beds between the wards is more balanced. In the original hospital case, the optimal fractional distribution was 43.2%, 32.4% and 24.3% for ward 1, 2 and 3, respectively. In Test 4 with $\Theta = 80$, this distribution changes to 42.5%, 31.3% and 26.3%. More importantly, as all wards receive more bed resources, the objective value is reduced correspondingly. Adding six additional beds yields a 38.9% reduction in the number of primary rejections.

Test 5 was conducted to assess the effect from relocation in the system on the optimal solution. To emphasize, we increased the demand for secondary hospitalizations in ward 3 substantially, by maximizing $p(f_1 = 0)_{13}$ and $p(f_2 = 0)_{23}$, keeping all other parameters fixed. Through these adjustments, we expect to increase the distance from initial to optimal solution. Moreover, we clarify how the optimal solution relates to a large probability of relocation within the system. Interestingly, it might seem natural to allocate beds to the ward with an increased demand (ward 3), however the optimal solution reveals the objective value is minimized by moving beds to ward 1 and 2, and avoiding relocation in the first place.

#	λ_1	λ_2	λ_3	Θ	p_{13}	p_{23}
1	**6.775**	3.96	2.52	74	0.23	0.27
2	5.42	**4.95**	2.52	74	0.23	0.27
3	5.42	3.96	**3.15**	74	0.23	0.27
4	5.42	3.96	2.52	**80**	0.23	0.27
5	5.42	3.96	2.52	74	**0.95**	**0.73**

Table 3.5: Test parameters used to assess Algorithm 6. Parameters subject to change are presented in bold font.

	Initial				**Optimal**							
#	M_1	M_2	M_3	$f(\boldsymbol{M})$	M_1^*	M_2^*	M_3^*	$f^*(\boldsymbol{M})$	Iter.	Eval.	A. Eval.	Runtime (s)
1	38	22	14	2.376	39	23	12	2.354	4	12	5	368.27
2	31	28	15	2.165	32	29	13	2.158	3	12	5	686.39
3	31	23	20	2.180	32	23	19	2.175	3	11	6	1757.23
4	33	25	22	1.106	34	25	21	1.103	2	11	6	3925.02
5	32	23	19	1.733	33	25	16	1.688	7	21	4	3253.00

Table 3.6: Results from the five parameter adjustment tests. Both initial and optimal solutions are presented. Information on the heuristic progression is presented in the last four columns.

Assessment of Expected Hospital Changes

For our last test, we consider a number of organizational changes planned to be introduced in the spring of 2016. Patients of another organizational region are to be rerouted to the case-hospital. As a result, patient arrival rate is expected to increase. Moreover, the case hospital are given additional resources to cope with the increase in demand, and for the area of gastrology, pneumology, endocrinology and geriatrics, available resources will increase from 74 to 93 beds. Patient arrival rate of type 1 and 2 are now expected at 9.84 and 3.44 patients per day, respectively.

Just as previously, we generate the initial solution, starting with $M^0 = (27\ 23)^T$. Rounding to the smallest estimated objective value, we set $M^* = (56\ 20)^T$ and initial objective value $f^* = 1.965$. After 4 iterations we find the new distribution of beds at $M_1 = 56$, $M_2 = 21$ and $M_3 = 16$, with an objective value of $f^* = 1.958$ patients per day. The total number of function evaluations is 9 with an overall runtime of **961.35** seconds.

3.5 Conclusion & Future Work

With a view to optimizing the distribution of bed resources, we presented a solution approach consisting of two main components. The first was a homogeneous continuous-time Markov chain (CTMC) used to evaluate the patient flow behavior. The second incorporated the Markov chain model in a heuristic to optimize the distribution of bed resources. For a specific hospital case, our

approach was used to find a 11.8% reduction in number of primary rejections – that is, the number of patients rejected on first arrival to the hospital. In addition, we found that a relatively small ($\approx 8\%$) increase in bed resources to the medical area has a potential to reduce this rejection rate from the current configuration with 38.9% fewer patients per day. Regarding this, hospital management should consider how the increase in resources relates to the overall cost, and if a potential increase in cost is compensated by the increased service level.

We collected data for the case-hospital by conducting interviews with hospital staff, and using patient data already registered in the hospital system. During this process we found dependencies in the flow system that stretches toward far more wards than were resources to include in this study. On the other hand, we found it reasonable to assume the medical area as an isolated system with patients going *out*, rather than *in* from other wards. Hourly arrival rate was found to be time-dependent, but with discharges mainly occuring during the day, the time-dependent behavior could be neglected, as was confirmed from observations of ward occupancy. Additionally, we found it reasonable to assume that patient length of stay was independent of the system load. However, for other applications where load-dependency cannot be neglected, such behavior can be implemented by defining service rates of the CTMC as function of the ward occupancy.

We statistically tested the CTMC model by replicating simulations of the CTMC itself. These were compared to hospital observations, and a simulated p-value of $p = 0.32$ was derived. We concluded that the CTMC model is not significantly different from the observed ward occupancy.

The local search heuristic was evaluated using a range of different tests. Firstly, the case-hospital result was checked by conducting a complete enumeration of the search space. Here, we found the heuristic solution was in fact the global optimal solution to the problem. However, as complete enumeration is foreseeable for this problem size, it is certainly not practical as a decision tool. Even though global optimality cannot be proven without, we propose to use our approach, with a 99.6% reduction in runtime. Secondly, we tested our local search heuristic conducting five tests with different parameter adjustments, and one additional test resembling a future organizational change. The local search heuristic performed well in all tests.

3.5.1 Future Work

For future work, a larger number of wards should be considered. We notice that such an expansion would require a substantial increase in state space, possibly reducing the practical use of our modeling approach. It should be considered how other methods could help to support the CTMC model approach with a view to decrease runtime spend on function evaluations. Moreover, as the problem complexity grows, other local search techniques, such as Tabu Search, might be more suitable approaches.

Lastly, to further support our modeling approach, simulation experiments should be conducted to assess the nature of the system under different parameter settings, as well as the CTMC robustness to different inter-arrival and service time distributions.

Acknowledgements

This research was supported by the Danish governmental organization Region Sjælland. The managing organization of seven public hospitals located on Zealand and Falster. We particularly thank the department of Production, Research and Innovation for providing us with the necessary data to conduct this research.

In addition, we would like to thank Senior Project Manager, Pernille Kirkvåg for providing us with essential information on patient flow, as well as Associate Prof. Anders Stockmarr for statistical advice.

Chapter 4

Strategic Room Type Allocation for Nursing Wards Through Markov Chain Modeling[1]

Anders Reenberg Andersen, Wim Vancroonenburg
and Greet Vanden Berghe

Abstract Providing patients with the best possible care is the most essential function of any hospital. In an increasing number of countries hospitals are governed by the number of patients they are able to attract and the corresponding services they provide for patients. One such service, which is often of significant importance for patients, is the option to choose their room type.

Hospital decision makers would benefit from a strategic method for optimizing the configuration of room types among nursing wards by distinguishing between patients who prefer private rooms and those who have no preference concerning whether they are assigned to a private or shared room. Such a decision support method is currently non-existent, therefore the goal of this study is to provide a methodology for hospital management. Specifically, a mixed modeling approach is proposed which evaluates the patient flow behavior by applying a Continuous-Time Markov Chain within a heuristic search procedure. This procedure recursively improves a configuration of rooms among the wards by sampling from a gradually improved interpolation of the objective function.

Based on patient data obtained from both a Danish and Belgian hospital, the performance and robustness of the proposed approach is validated through various numerical experiments, demonstrating that solutions within a relative gap of 1% from the optimum are attained in most cases.

4.1 Introduction

Rising public expenditure concerning health care systems has led many governments to apply budgetary pressure on hospitals to rationalize their spending [45]. At the same time, competition in hospital services is employed in many countries as a mechanism to motivate hospitals to reduce costs in order

[1] Submitted to Artificial Intelligence in Medicine

to remain competitive [95]. Since patients may be offered freedom-of-choice regarding their hospital admission, hospitals therefore compete against one another to attract more patients. Not only do they compete on the basis of the medical services they provide, but also with regard to amenities that increase patient comfort during their stay. One such service of importance is the option for admitted patients (also referred to as *inpatients*) to choose their preferred room *type*. Many different room types may be distinguished in nursing wards, varying in capacity (examples include ward room, double room and private room) and amenities (in maternity wards some rooms may provide a shower or an extra bed for a spouse). There are significant financial incentives for hospitals to meet patients' room type preferences: for example, hospitals in Belgium may charge room supplements when meeting private room demands and physicians may even charge honorarium supplements (hospital bills may be up to five times more expensive for private rooms than for shared) if such preferences are met [89]. However, when a patient does not have such a room preference, but is still admitted to a private room due to lack of room availability of different cheaper room types, such supplements cannot be charged. A survey in Belgian hospitals by Verhelst, 2009 [134] further shows how preferred room type unavailability may even be a cause for postponing admissions. It is therefore of considerable importance for hospital administrators to address these concerns by matching the availability of different room types with the respective demand by patients in order to maximize revenue.

This study focuses on the decision problem of hospital administrators who wish to address this issue by reallocating existing room infrastructure between different nursing wards belonging to different hospital units such as different surgical disciplines. These units may have different patient arrival patterns, length-of-stay (LOS) distributions and room preference profiles that, for historical or organizational reasons, do not match their currently-allocated infrastructure. Hence, reallocation may be necessary to match current patient preferences. Currently, a methodology for finding a suitable reallocation is non-existent.

To this end, an approach is presented which accounts for the patient flow behavior using a Continuous-Time Markov Chain (CTMC) model. This model assesses the allocation of rooms in a heuristic search procedure, where the solution is gradually improved by sampling randomly from an interpolation of the objective function. Based on hospital data obtained from both a Danish and Belgian hospital, the performance of the proposed approach is validated with a range of numerical experiments.

The remainder of this paper is organized as follows. First, the present work is positioned in the context of the relevant literature in Section 4.1.1. In Section 4.2, the specific assumptions and problem details of this study are elaborated upon. Section 4.3 present the proposed solution approach followed by Section 4.4 which applies the approach by way of computational experimentation. Lastly, in Section 4.5 conclusions and future research directions are presented.

4.1.1 Literature Review

The number of operations research methods that specifically address room capacity optimization in nursing wards, as opposed to solely considering bed capacity, is limited. Those studies that do consider this specific aspect generally concern decision problems at an operational decision level, where such details regarding room infrastructure cannot be ignored. Notably, Demeester et al., 2010 [47] formulated and studied a patient admission scheduling problem that addresses the assignment of admitted patients to beds over a given, short-term, planning horizon, considering room type and equipment. The consideration of gender conflicts in shared rooms and room type preferences by patients requires explicit modeling of room infrastructure. Given that Demeester et al., 2010 formulated a challenging combinatorial optimization problem along with problem instances, they triggered a series of different studies further investigating algorithm development [23, 33, 107], different modeling aspects [34, 131, 35], and complexity [132]; all of which include the explicit consideration of room infrastructure. Most of these studies apply metaheuristic optimization techniques to deal with problems of realistic size. Nevertheless, three studies apply Mixed Integer Linear Programming (MILP) to models of reduced size [132], in a dynamic setting [131] (where sub-problems are typically smaller) or combined with column generation [107] to improve lower and upper bounds. Other studies in this area, though not derived from the formulation of Demeester et al., 2010, demonstrate that considering room infrastructure is necessary for the practical implementation of systems. Bachouch et al., 2012 [16] presented a hospital bed management problem where patient admissions are scheduled while considering no-mixed gender rooms, isolation of contagious patients in single rooms or alone in double rooms, and incompatibilities between pathologies. A MILP model is formulated which is subsequently applied to different solvers in a computational comparative study. Schmidt et al., 2013 [111] also presented a decision support model for admission planning and assignment to wards. Their model also explicitly accounts for the availability of beds in either private or shared rooms, depending on the planned patients' preferences. Both an exact approach, using a MILP formulation, and heuristic approaches compared in a computational study.

The application of Markov Chains to model patient flow is an uncommon approach compared to, for example, simulation-based modeling methods [22, 85]. Nonetheless, Markov Chains have been successfully applied in a variety of different cases in the last few years. Bartolomeo et al., 2008 [19] applied a Discrete-Time Markov Chain (DTMC) model to assess the readmission probability of patients. Further, Broyles et al., 2010 [29] applied a DTMC to predict the number of inpatients, demonstrating how their model attains superior predictability compared to a seasonal Autoregressive Integrated Moving Average model.

Concerning CTMCs, both He et al., 2017 [68] and Shao et al., 2013, [115] developed a model to assess and identify bottlenecks with regard to the colonoscopy screening process and surgical operations in an emergency de-

partment, respectively. Furthermore, Wang et al., 2014 [135] apply a CTMC to model care delivery to patients in rooms by modeling the system as a closed network. Lastly, Shaw & Marshall, 2007 [116] evaluate the LOS for heart-failure patients, demonstrating how the Coxian phase-type distribution is adequate in this regard. The focus of all these studies is on modeling patient flow, while none of them utilize their approach to optimize the system. Only Andersen et al., 2017 [13] model patient hospitalization and relocation to multiple wards and employ a heuristic to optimize ward capacity. However, their study only considers capacity on an aggregated level and does not account for room infrastructure.

4.1.2 Contribution

Interestingly, most studies considering room infrastructure availability have done so only at an operational decision level, where this aspect cannot be ignored. However, matching infrastructure availability to demand is of strategic importance for hospitals in the context of maximizing revenue and providing enhanced service to patients. Currently, no existing methodology for strategic room/bed (re)allocation considers the aspect of room types. This study proposes a CTMC model combined with a heuristic search procedure to address this aspect at the strategic decision level, where capacity and infrastructure can be reallocated between hospital units to better match individual demand patterns. To our knowledge, this is the first analytical approach that accounts for patient arrivals, relocation, and room type preferences; and, furthermore, where room configuration is optimized.

4.2 Problem Description

The decision problem studied in this paper involves the allocation of room types to nursing units of different medical specialisms. In this setting, the most differentiating characteristic between room types, namely being either private (one bed per room) or shared (two or more beds per room), is scrutinized. It is assumed that the total availability of private and shared rooms is fixed, but that room types may be reassigned between units. Such situations may occur, for example, when different nursing units occupying a single, physical area are reorganized or when patient characteristics such as LOS distributions or private room preferences have changed, necessitating a reallocation in order to realign available room infrastructure with demand.

Patients are assumed to arrive at the hospital according to a time-homogeneous process where both inter-arrival time and LOS are random. Furthermore, patients can be grouped into types such that each type prefers admission to a specific nursing ward. However, when capacity is insufficient, patients must not be made to wait for a bed, but be relocated to a ward where capacity is available. In addition, a certain fraction of the patients prefer admission to a private room, whereas the remaining patients have no preference concerning

whether their room is private or shared. These assumptions are elaborated upon at greater length later in Section 4.3.2.

Formal Definition

Formally, a hospital setting is considered which features a set of wards $\mathcal{W}$, $|\mathcal{W}|$ types of patients, with each type preferring admission to a unique ward $i \in \mathcal{W}$, and a set of room types $\mathcal{R}$. Let $u_{ir} \in \mathbb{N}_0$ define the number of rooms of type $r \in \mathcal{R}$ that have been allocated to ward $i \in \mathcal{W}$ and $b_r \in \mathbb{N}_0$ define the capacity associated with each room type. Further, let set $\mathcal{R}$ feature a subset of room types for which $b_r > 1$, and a *private* room type where $b_r = 1$. Finally, let $N_r \in \mathbb{N}_0$ define the available number of rooms of type $r \in \mathcal{R}$, and $M_i \in \mathbb{N}_0$ define the aggregated capacity of each ward $i \in \mathcal{W}$. Then,

$$\sum_{i \in \mathcal{W}} u_{ir} = N_r \qquad \forall r \in \mathcal{R} \tag{4.1}$$

and,

$$\sum_{r \in \mathcal{R}} u_{ir} b_r = M_i \qquad \forall i \in \mathcal{W} \tag{4.2}$$

Now, let $\mathbf{u}$ define a matrix of the elements $u_{ir}\ \forall i \in \mathcal{W}, r \in \mathcal{R}$, and consider that:

- $f(\mathbf{u})$ yields the expected total number of patients relocated to an alternative ward per day, an alternative ward being defined as a ward having spare capacity.
- $g(\mathbf{u})$ yields the expected total number of patients who prefer a private room *and* are correspondingly assigned to one.

Let $\tau \in \mathbb{R}_{>0}$ denote an upper bound on $f(\mathbf{u})$ ensuring that a substantial number of patients will receive their preferred care. The objective of this study is therefore to derive a configuration of the room types, u_{ir}, that fulfills,

$$\text{Maximize} \qquad g(\mathbf{u}) \tag{4.3}$$

$$\begin{aligned}
\text{Subject to} \qquad & f(\mathbf{u}) \leq \tau && && (4.4)\\
& \textstyle\sum_{i \in \mathcal{W}} u_{ir} = N_r && \forall r \in \mathcal{R} && (4.5)\\
& \textstyle\sum_{r \in \mathcal{R}} u_{ir} b_r \geq 1 && \forall i \in \mathcal{W} && (4.6)\\
& u_{ir} \in \mathbb{N}_0 \quad \forall i, r \in \mathcal{W}, \mathcal{R} && && (4.7)
\end{aligned}$$

The aim of formulation (4.3)-(4.7) is to attain the maximum expected number of patient-room preference matches, subject to a limited number of relocated patients (Constraint 4.4), a fixed capacity of each room type (Constraint 4.5) and an assignment of minimum one room per ward (Constraint 4.6).

4.3 Modeling & Solution Approach

The evaluation of $g(\mathbf{u})$ and $f(\mathbf{u})$ depends on an explicit modeling of the admission process' related queueing system that arises from the random arrival and room occupation of patients in nursing wards. Due to the complexity of this queueing system (cf. Section 4.3.2), optimization problem (4.3)-(4.7) cannot be solved to optimum without complete enumeration. Therefore, this study proposes a heuristic search procedure.

The Randomized & Interpolated Search (RIS) heuristic proposed in this study applies an iterative procedure to sample good solutions from the solution space of (4.3)-(4.7), where in each iteration a new solution is selected based on an interpolation of scattered samples from the solution space. The overall structure of this approach is detailed in Section 4.3.1. To evaluate the behavior of the queueing system associated to each solution, a time-homogeneous CTMC model proposed by Andersen et al., 2017 [13] is employed to derive the expected room occupancy. The CTMC model will be presented in Section 4.3.2. Since the CTMC is computationally expensive, Section 4.3.3 presents a core element of our heuristic search procedure, an approximative, fast, surrogate objective function. Finally, given that patient behavior is considered to be exclusively dependent upon aggregated ward capacity, the *room* configuration can be derived with Integer Linear Programming (ILP). The precise means by which this is achieved is detailed throughout Section 4.3.4

4.3.1 Randomized & Interpolated Search (RIS) heuristic

Consider a given room configuration $\mathbf{u}$ and recall capacity constraints (4.5) and (4.6). Now, consider the solution space U resulting from these constraints, and let $Y_f(\mathbf{u})$ and $Y_g(\mathbf{u})$ yield an estimate of $f(\mathbf{u})$ and $g(\mathbf{u})$ based on an interpolation of some known solutions in this space, respectively. Let $\mathbf{x}$ define the set of these known solutions and $Z(\mathbf{u})$ define a Probability Density Function (PDF) that corresponds proportionally to $Y_g(\mathbf{u})$ and sums to unity. Then, in order to approach the configuration of rooms that attains the maximum of $g(\mathbf{u})$, the following stepwise procedure is considered:

1. Select a range of initial solutions for $\mathbf{x}$.
2. Calculate $g(\mathbf{u})$ and $f(\mathbf{u})$ based on $\mathbf{x}$.
3. Derive $Y_f(\mathbf{u})$ and $Y_g(\mathbf{u})$ by interpolating between the known solutions in $\mathbf{x}$.
4. Derive $Z(\mathbf{u})$ in accordance with $Y_g(\mathbf{u})$.
5. Add a new configuration, $\mathbf{u}'$, to $\mathbf{x}$ by sampling from PDF $Z(\mathbf{u})$, constrained by $Y_f(\mathbf{u}) \leq \tau$ in accordance with (4.4), and calculate $g(\mathbf{u}')$ and $f(\mathbf{u}')$.
6. If the elapsed time exceeds the fixed time limit then **stop**; otherwise return to step **3**.

The procedure is initialized by requiring that $\mathbf{x}$ contains the $|\mathcal{W}|$ extreme points in which all room types have been moved to a single ward, respecting lower capacity bound (4.6), and thus ensuring that all room configurations are included in the interpolation. Next, $\mathbf{x}$ is expanded and a basis for the interpolation is created by sampling uniformly from U.

By applying this procedure, $Y_g(\mathbf{u})$ recursively approaches $g(\mathbf{u})$ in the space constrained by (4.4)-(4.7). Notice how the interpolations $Y_g(\mathbf{u})$ and $Y_f(\mathbf{u})$ are gradually improved based on the solution samples $\mathbf{x}$ where $Y_g(\mathbf{u})$ is employed to focus the search through PDF $Z(\mathbf{u})$ and $Y_f(\mathbf{u})$ is employed to estimate the feasible space.

When the sampling based on $Y_g(\mathbf{u})$ is rather widespread, the probability mass is concentrated upon the promising regions by performing the conversion $\tilde{Y}_g(\mathbf{u}) = Y_g(\mathbf{u})^\beta$, thereby amplifying the curvature of the interpolation. However, this still requires an initialization of $Y_g(\mathbf{u})$ based on uniformly distributed solution-evaluations throughout U. In other words, runtime is potentially wasted in regions that are not relevant to the objective. To overcome this, let $\tilde{f}(\mathbf{u})$ and $\tilde{g}(\mathbf{u})$ define surrogates of functions $f(\mathbf{u})$ and $g(\mathbf{u})$ that have similar optima, but shorter evaluation times. Thus, by conducting the initialization using the surrogate $\tilde{g}(\mathbf{u})$ and then switching to the true objective function, $g(\mathbf{u})$, for the remaining steps, the *true*, and slower, solution-evaluations are *only* performed in the most promising region of the search space.

Let $\tilde{\mathbf{x}}$ define the set of configurations that have been evaluated using the aforementioned surrogate function. Then, as the search procedure progresses, the interpolation will be derived on the basis of $\tilde{\mathbf{x}}$ as well as the gradually increasing set $\mathbf{x}$. Now, to ensure that $\mathbf{x}$ can replace $\tilde{\mathbf{x}}$ in a limited number of iterations, a proximity tolerance ξ is defined such that if the euclidean distance $\sqrt{\sum_{i\in\mathcal{W}}(M_i - \tilde{M}_i)^2}$ is smaller than or equal to ξ, where M_i and $\tilde{M}_i$ is the aggregated capacity of an element in $\mathbf{x}$ and $\tilde{\mathbf{x}}$, respectively, then the surrogate solution associated with $\tilde{M}_i$ is removed from $\tilde{\mathbf{x}}$. The final search procedure is documented in Algorithm 7. The implications of varying β and ξ are elaborated upon in Section 4.4.

4.3.2 Evaluating $g(\mathbf{u})$ and $f(\mathbf{u})$

Recall the system presented in Section 4.2, featuring a set of patient types requiring assignment to a set of wards, where in case of insufficient capacity patients are either moved to an alternative ward or admitted to a location that is not included in the model, i.e. lost from the system. To model this behavior, a CTMC approach [13] is employed.

Consider a time-homogeneous CTMC with state space S and state definition $s = (w_{11}, w_{21}, \cdots, w_{ij}, \cdots, w_{|\mathcal{W}||\mathcal{W}|})$, where w_{ij} is the number of type i patients hospitalized in ward j, with $i, j \in \{1, 2, \cdots, |\mathcal{W}|\}$. Let f_j define the number of free beds in ward j, so $f_j = M_j - \sum_{i\in I} w_{ij}$, where M_j is the ag-

Algorithm 7 The RIS heuristic.

$\tilde{\mathbf{x}} \leftarrow uniformSampling()$ ▷ Initialize and evaluate surrogate sampling
$\mathbf{x} \leftarrow \emptyset$
while $elapsedTime < timeLimit$ **do**
  $Y \leftarrow interpolate(\tilde{\mathbf{x}}, \mathbf{x})$
  $\tilde{Y} \leftarrow exponentiate(Y)$ ▷ Exponentiate the interpolation
  $Z \leftarrow convertToPDF(\tilde{Y})$
  $\mathbf{x} \leftarrow addNewSample(Z, \mathbf{x})$ ▷ Add and evaluate new sample
  $\tilde{\mathbf{x}} \leftarrow remove(\mathbf{x}, \tilde{\mathbf{x}})$ ▷ Check and remove proximate surrogate samples
end while
$\mathbf{u}^* \leftarrow getBest(\mathbf{x})$
return $\mathbf{u}^*$

gregated capacity of each ward. Additionally, let λ_i define the arrival rate of patient type i and that all patient arrivals, regardless of their type, are generated according to a Poisson process. Moreover, let μ_i define the service rate of patient type i, assuming that inter-service times are exponentially distributed. Furthermore, let $p(f_1, f_2, \cdots, f_{|\mathcal{W}|})_{ij}$ define the fraction type i patients hospitalized in ward j, governed by the number of free beds in each ward; f_1, f_2, ... and $f_{|\mathcal{W}|}$.

Let q_{ss^*} define the rate at which the system transitions from a current state $s \in S$ to a new state $s^* \in S$. Then,

$$q_{ss^*} = \begin{cases} \lambda_i & \text{if } s^* = (\cdots, w_{ii}+1, \cdots, f_i - 1, \cdots) \text{ and } f_i > 0 \quad \forall i \in I \\ \lambda_i p(f_i = 0)_{ij} & \text{if } s^* = (\cdots, w_{ij}+1, \cdots, f_j - 1, \cdots) \text{ and } f_i = 0, f_j > 0 \quad \forall i, j \in I, i \neq j \\ \lambda_i p(f_i = 0, f_k = 0)_{ij} & \text{if } s^* = (\cdots, w_{ij}+1, \cdots, f_j - 1, \cdots) \text{ and } f_i = 0, f_k = 0, f_j > 0 \quad \forall i, j, k \in I, i \neq j \neq k \\ \vdots & \quad \vdots \\ \lambda_i p(f_i = 0, f_k = 0, \cdots, f_l = 0)_{ij} & \text{if } s^* = (\cdots, w_{ij}+1, \cdots, f_j - 1, \cdots) \text{ and } f_i = 0, f_k = 0, \cdots, f_l = 0, f_j > 0 \\ & \forall i, j, k \cdots l \in I, i \neq j \neq k \neq \cdots \neq l \\ \mu_i w_{ij} & \text{if } s^* = (\cdots, w_{ij}-1, \cdots, f_j + 1, \cdots) \text{ and } w_{ij} > 0 \quad \forall i, j \in I \end{cases}$$

where $p(f_i = 0, f_k = 0, f_j > 0, \cdots, f_N > 0)$ is abbreviated $p(f_i = 0, f_k = 0)$, $s^* = (\cdots, w_{ij} + 1, \cdots, f_j - 1, \cdots)$ indicates the arrival of patient i to a ward j, and $s^* = (\cdots, w_{ij} - 1, \cdots, f_j + 1, \cdots)$ a corresponding discharge. Let Q define the transition rate matrix of rates q_{ss^*} $\forall s, s^* \in S$. To derive state distribution π, S is truncated and the global balance equations $\pi Q = 0$ are solved using successive overrelaxation [123, p. 311] with a relaxation parameter equal to 1.75 and convergence measured on the largest relative difference between successive iterations in accordance with Andersen et al., 2017 [13].

Let $\pi_i(n)$ define the probability of exactly $n \in \mathbb{N}_0$ patients being hospitalized in ward $i \in \mathcal{W}$; $0 \leq n \leq M_i$. $\pi_i(n)$ is consequently a marginal distribution to the state distribution of the CTMC. Recall that all patients may be categorized as one of two types: patients who prefer a private room, X, and patients that have no preference concerning whether they are assigned to a shared or

private room. In general, the probability that a patient prefers a private bed is $\psi \in \mathbb{R}_{0 \leq \psi \leq 1}$. Now, let $P_i(x, y)$ define the probability that exactly x patients preferring a private bed, and $n - x = y$ patients who do not care whether their room is shared or private, are hospitalized in ward $i \in \mathcal{W}$. Then,

$$P_i(x, y) = b(x; x + y, \psi) \cdot \pi_i(x + y) \tag{4.8}$$

where $b(x; n, \psi) = Prob(X = x)$ is the probability mass function of the binomial distribution with $n = x + y$ trials, and success probability ψ. Further, let $\rho_i(x)$ define the probability that exactly x beds are occupied by patients who prefer a private bed. Then,

$$\rho_i(x) = \sum_{y=0}^{M_i - x} P_i(x, y) \tag{4.9}$$

which from (4.8) results in:

$$\rho_i(x) = \sum_{y=0}^{M_i - x} \Big(b(x; x + y, \psi) \cdot \pi_i(x + y) \Big) \tag{4.10}$$

Function $\rho_i(x)$ is essential to both the definition of $f(\mathbf{u})$ and $g(\mathbf{u})$, as will be demonstrated in what follows.

Assume the following ordering of patients as they are hospitalized:

1. Whenever patients who prefer private beds are hospitalized, they will *always* be assigned to a private room if one is available.

2. Patients who have no preference regarding room types, are *only* assigned to a private room if no shared room capacity is available.

By observing an arbitrary ward $i \in \mathcal{W}$, the expected number of patients who prefer a private room and are correspondingly assigned to one is $\sum_{x=0}^{u_{i,private}} (x \cdot \rho_i(x)) + \sum_{x=u_{i,private}+1}^{M_i} (u_{i,private} \cdot \rho_i(x))$, resulting in the following objective function:

$$g(\mathbf{u}) = \sum_{i \in \mathcal{W}} \left(\sum_{x=0}^{u_{i,private}} (x \cdot \rho_i(x)) + \sum_{x=u_{i,private}+1}^{M_i} (u_{i,private} \cdot \rho_i(x)) \right) \tag{4.11}$$

where $u_{i,private}$ is the number of private rooms allocated to ward $i \in \mathcal{W}$. Notice that $g(\mathbf{u})$ is, in essence, independent of the characteristics of the shared room types. Regarding $f(\mathbf{u})$, which ensures an upper bound on the number of relocated patients through (4.4), the overall flow of patients into the hospital is of more concern. Consider blocking probability $\pi_i^B = \pi_i(M_i)$ of ward $i \in \mathcal{W}$, then

$$f(\mathbf{u}) = \sum_{i \in \mathcal{W}} \lambda_i \pi_i^B \tag{4.12}$$

denotes the total expected number of patients who are rejected and correspondingly relocated upon arrival. Figure 4.1 depicts the dependencies between the CTMC and Expressions (4.11) and (4.12), respectively. Notice that the behavior of the system, as evaluated by the CTMC, depends only on the aggregated capacity. This feature will be exploited in the search procedure using ILP modeling introduced in Section 4.3.4.

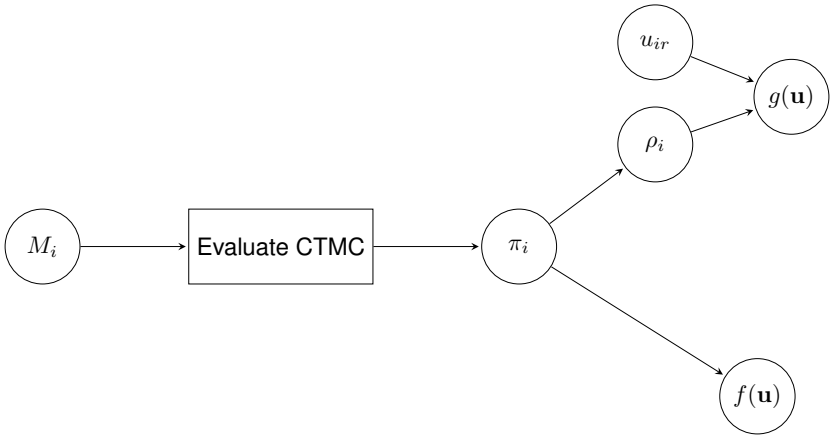


Figure 4.1: Dependencies in evaluating $g(\mathbf{u})$ and $f(\mathbf{u})$.

4.3.3 The Surrogate Functions

Consider the flow of patients into a single ward, as depicted in Figure 4.2. From this perspective one notices that the system approaches an $M/M/c/c$ queueing model as the number of arriving relocated patients decrease. That is, a queue where the capacity of the entire system equals the number of servers. For the $M/M/c/c$ model,

$$\tilde{\pi}_i(n) = \frac{(\lambda_i/\mu_i)^n/n!}{\sum_{k=0}^{M_i}(\lambda_i/\mu_i)^k/k!} \tag{4.13}$$

where $\tilde{\pi}_i(n)$ is the probability that exactly $n \in \mathbb{N}_0$ patients are hospitalized in ward $i \in \mathcal{W}$ [123, p. 434]. Equation (4.13) is therefore an approximation of $\pi_i(n)$, which accuracy decreases as more patients are relocated within the system. Correspondingly, if all patients are lost from the system on arrival, then Equation (4.13) substitutes for $\pi_i(n)$ exactly. $f(\mathbf{u})$ may therefore be approximated by:

$$\tilde{f}(\mathbf{u}) = \sum_{i \in \mathcal{W}} \lambda_i \tilde{\pi}_i^B \tag{4.14}$$

where $\tilde{\pi}_i^B = \tilde{\pi}_i(M_i)$. Similarly, in the surrogate for $g(\mathbf{u})$, $\tilde{\pi}_i(n)$ is employed to approximate (4.10) by,

$$\tilde{\rho}_i(x) = \sum_{y=0}^{M_i - x} \Big(b(x; x+y, \psi) \cdot \tilde{\pi}_i(x+y) \Big) \tag{4.15}$$

which is then used to substitute $\rho_i(x)$ in Equation (4.11), leading to $\tilde{g}(\mathbf{u})$. Notice that when (4.13) replaces $\pi_i(n)$ from the CTMC, the computational effort of setting-up and applying successive over-relaxation is avoided which is the proposed approach to the global balance equations, $\pi Q = 0$ [13]. As a result, the search procedure is scoped rather quickly by creating an initial outline of both $f(\mathbf{u})$ and $g(\mathbf{u})$.

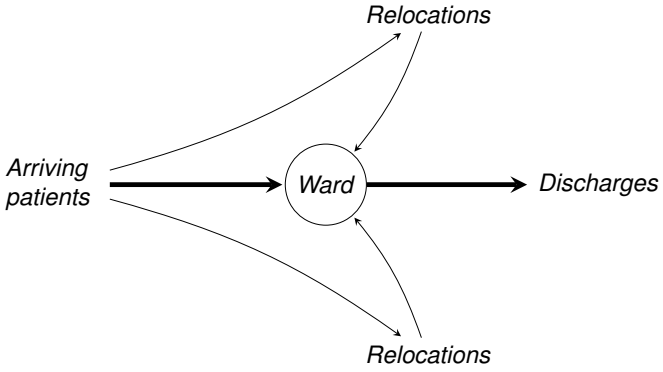


Figure 4.2: Flow of patients *to* and *from* a single ward.

4.3.4 Sub-Optimal Room Configuration

Recall the dependencies in deriving functions $f(\mathbf{u})$ and $g(\mathbf{u})$, depicted in Figure 4.1. We only require the aggregated capacity M_i to evaluate the system through the CTMC. Not until then is the room configuration $\mathbf{u}$ applied to the patient occupancy distribution ρ_i to derive the objective value $g(\mathbf{u})$. Hence, by assuming a fixed aggregated capacity the problem of maximizing $g(\mathbf{u})$ reduces to the following ILP model:

$$\text{Maximize} \quad \sum_{i \in \mathcal{W}} \sum_{j \in \mathcal{J}_i} x_{ij} w_{ij} \tag{4.16}$$

$$\begin{aligned}
\text{Subject to} \quad & \sum_{j \in \mathcal{J}_i} x_{ij} = 1 && \forall i \in \mathcal{W} && (4.17)\\
& \sum_{j \in \mathcal{J}_i} x_{ij} \cdot j + \sum_{r \in \mathcal{R}'} y_{ir} b_r = M_i^* && \forall i \in \mathcal{W} && (4.18)\\
& \sum_{i \in \mathcal{W}} \sum_{j \in J_i} x_{ij} \cdot j \leq N_{private} && && (4.19)\\
& \sum_{i \in \mathcal{W}} y_{ir} b_r \leq N_r && \forall r \in \mathcal{R}' && (4.20)\\
& x_{ij} \in \{0, 1\} && \forall i, j \in \mathcal{W}, \mathcal{J}_i && (4.21)\\
& y_{ir} \in \mathbb{N}_0 && \forall i, r \in \mathcal{W}, \mathcal{R}' && (4.22)
\end{aligned}$$

Let $\mathcal{R}'$ define the set of shared room types. That is, $\mathcal{R}' \subset \mathcal{R}$ and $|\mathcal{R}'| = |\mathcal{R}| - 1$. Further, let set $\mathcal{J}_i = \{0, 1, 2, \ldots, M_i\}$ account for the number of beds that can be assigned to private rooms in each ward $i \in \mathcal{W}$.

Additionally, let the decision variable $x_{ij} \in \{0, 1\}$ equal 1 whenever ward $i \in \mathcal{W}$ is assigned j private beds, where $j \in \mathcal{J}_i$; and otherwise 0. Further, let parameter $w_{ij} \in \mathbb{R}_{\geq 0}$ define the expected number of patients who both prefer *and* are also assigned to private beds in ward $i \in \mathcal{W}$, given that j private beds are available in this ward. That is, following the convention in Equation (4.11),

$$w_{ij} = \sum_{k=0}^{j} \big(k \cdot \rho_i(k)\big) + \sum_{k=j+1}^{M_i} \big(j \cdot \rho_i(k)\big) \tag{4.23}$$

where as before $\rho_i(k)$ is derived using Equation (4.10), resulting in the objective function $\sum_{i \in \mathcal{W}} \sum_{j \in \mathcal{J}_i} x_{ij} w_{ij}$, which yields exactly the same result as Equation (4.11).

Lastly, parameter $y_{ir} \in \mathbb{N}_0$ defines the number of shared room types $r \in \mathcal{R}'$ assigned to ward $i \in \mathcal{W}$.

Constraints (4.17)-(4.20) are defined as follows:

- (4.17) ensures that each ward receives a fixed amount of private beds.
- (4.18) ensures that the distribution of the aggregated capacity is maintained, keeping the parameter w_{ij} valid.
- Finally, (4.19) and (4.20) restrict the maximum occurrence of each room type. Recall that N_r defines the total number of room types $r \in \mathcal{R}$ available to the hospital.

By evaluating $g(\mathbf{u})$ and $f(\mathbf{u})$, using the aforementioned CTMC, ILP formulation (4.16)-(4.22) can be employed to yield the optimum room configuration conditioned by the distribution of the aggregated capacity. Therefore, instead of evaluating based on the room configuration directly, this feature is exploited in our heuristic search procedure by applying aggregated capacity M_i as the decision variable.

Recall Algorithm 7, where $\mathbf{x}$ and $\tilde{\mathbf{x}}$ contain the samples for which the true and surrogate functions have been evaluated, respectively. By adding the aggregated capacity to $\mathbf{x}$ and $\tilde{\mathbf{x}}$, the associated room configuration is derived as follows:

1. Evaluate $g(\mathbf{u})$ and $f(\mathbf{u})$ as per Sections 4.3.2 and 4.3.3.
2. Derive the sub-optimal room configuration by ILP formulation (4.16)-(4.22).

4.4 Numerical Study

In this section, the RIS heuristic presented in Section 4.3 is evaluated in a range of numerical experiments to assess its performance. These experiments are conducted on hospital data introduced by Andersen et al., 2017 [13], and room infrastructure data from a Belgian hospital. All experiments are implemented in Java, including the CTMC from Section 4.3.2. To derive $Y_g(\mathbf{u})$ and $Y_f(\mathbf{u})$ natural neighbor interpolation [117] is employed using the `SibsonInterpolator2` class of the Java Mines Toolkit[2]. Lastly, the ILP model presented in Section 4.3.4 is solved using IBM ILOG CPLEX 12.7.1.

4.4.1 Data Description

The data for our subsequent experiments is based on three different datasets which are obtained from the study by Andersen et al., 2017 [13]. The data accounts for three different wards and consists of patient arrival rates and length of stay distributions; the respective routing probabilities in the system, and lastly the total bed capacity. No data was obtained specifically with regard to the number of room types for this case. However, data from a Belgian hospital[3] suggests that the number of private rooms may easily constitute half of the total bed capacity. This proportion will serve as the basis for the three sets. Furthermore, even though the presented approach may be generalized to any capacity configuration for the shared room types, only a single shared room type consisting of two beds is considered, next to a single private room type (i.e. $|\mathcal{R}| = 2$).

All experiments primarily consider a dataset referred to as the *original* set, which is based solely on true patient data. Two additional sets, *high arrival rate* and *high relocation*, are derived from the original data by adjusting the arrival rate and routing probability parameters, respectively. These additional sets are included to assess the potential changes in patient characteristics. In addition, since no data was obtained concerning the proportion of private patients, a value of $\psi = 0.2$ is assumed, unless otherwise stated.

The parameters associated with each dataset are presented in Tables 4.1 and 4.2. Furthermore, the initialization of the RIS heuristic includes a minimization of the expected number of relocated patients, $f(\mathbf{u})$ from Andersen et al., 2017 [13]. Each minimization, denoted as $\min\{f(\mathbf{u})\}$, is presented in Table 4.3.

[2]Java Mines Toolkit on interpolation and gridding - `http://dhale.github.io/jtk/api/edu/mines/jtk/interp/package-summary.html`

[3]Data supplied by hospital AZ Maria Middelares, based in Gent, Belgium, in the context of iMinds ICON project AORTA - `https://www.imec-int.com/nl/imec-icon/research-portfolio/aorta`

Dataset	λ_1	λ_2	λ_3	μ_1	μ_2	μ_3	Total Beds	$N_{private}$	N_{double}
Original	5.42	3.96	2.52	0.19	0.19	0.11	74	36	19
High Arrival Rate	6.78	3.96	2.52	0.19	0.19	0.11	74	36	19
High Relocation	5.42	3.96	2.52	0.19	0.19	0.11	74	36	19

Table 4.1: Rates and capacities associated with each of the three datasets. All rates and the total bed capacity are obtained from Andersen et al., 2017 [13], whereas the ratio of private to shared rooms, $\frac{N_{private}}{N_{double}}$ is based on data from a Belgian hospital[3].

Dataset	p_{11}	p_{12}	p_{13}	p_{21}	p_{22}	p_{23}	p_{31}	p_{32}	p_{33}
Original	-	0.05	0.23	0.10	-	0.27	0.06	0.00	-
High Arrival Rate	-	0.05	0.23	0.10	-	0.27	0.06	0.00	-
High Relocation	-	0.05	0.95	0.10	-	0.90	0.06	0.00	-

Table 4.2: The routing probabilities associated with each of the three datasets, respectively. All parameter values have been obtained from Andersen et al., 2017 [13].

Dataset	$\min\{f(\mathbf{u})\}$	M_1	M_2	M_3
Original	1.592	32	24	18
High Arrival Rate	2.354	39	23	12
High Relocation	1.688	33	25	16

Table 4.3: Each minimization, $\min\{f(\mathbf{u})\}$, obtained from Andersen et al., 2017 [13]. The associated distribution of beds for each dataset is provided.

4.4.2 Error of the Surrogate Function

Prior to evaluating the heuristic search procedure, an assessment of the error of both surrogate functions was performed by conducting a full enumeration of the search space. In order to accommodate this, room availability was limited to $N_{private} = 20$ private and $N_{double} = 10$ shared double rooms for the *high arrival rate* and *high relocation* datasets. Otherwise, the full availability of rooms for the *original* dataset was employed. The enumeration was conducted using the parameters from all three datasets (Table 4.1 and 4.2) on both the *true* functions $g(\mathbf{u})$ and $f(\mathbf{u})$, and *surrogate* functions $\tilde{g}(\mathbf{u})$ and $\tilde{f}(\mathbf{u})$.

Results were evaluated by calculating the error and comparing each functions' optimum. Table 4.4 presents the euclidean distance between the optima of $g(\mathbf{u})$ and $\tilde{g}(\mathbf{u})$. Notice that this is measured on the distribution of the aggregated capacity given how this is the primary decision variable. Table 4.4 includes the relative error concerning the optimum of the surrogate function. Lastly, Figure 4.3 shows the error of the original data, namely $g(\mathbf{u}) - \tilde{g}(\mathbf{u})$. The figure also illustrates the optima of both the true and surrogate objective function.

Recall from Section 4.4.2 that the surrogate functions are based on $M/M/c/c$ model (4.13) which does not account for the hospitalization of relocated pa-

tients. It is therefore expected that the surrogate functions will lose accuracy as the number of relocated patients increases. That is, either when general routing probabilities are high, or if the wards lack capacity. The latter situation is reflected in Figure 4.3, showing that the error is smaller when the capacity is more evenly distributed, thereby resulting in fewer relocated patients.

Regarding the routing probabilities, Table 4.4 demonstrates how the relative error is fairly robust with regard to the changes between the *original* and *high relocation* datasets. However, the optima have changed substantially from a euclidean distance of approximately 1.4 to 20.8 beds; therefore, demonstrating that when a substantial number of patients are relocated within the system, one must rely on the RIS heuristic being able to adapt interpolation $Y_g(\mathbf{u})$ to objective function $g(\mathbf{u})$, despite the accuracy of the surrogate function.

Dataset	$\sqrt{\sum_{i\in\mathcal{W}}(M_i^* - \tilde{M}_i^*)^2}$	$(g(\tilde{\mathbf{u}}^g)/\tilde{g}(\tilde{\mathbf{u}}^g)) - 1$	$(f(\tilde{\mathbf{u}}^f)/\tilde{f}(\tilde{\mathbf{u}}^f)) - 1$
Original	1.414	0.016	0.092
High Arrival Rate	6.164	0.010	0.050
High Relocation	20.833	0.018	0.053

Table 4.4: The euclidean distance between the optima of $g(\mathbf{u})$ and $\tilde{g}(\mathbf{u})$, and the relative error at the optimum, $\tilde{\mathbf{u}}^g$ and $\tilde{\mathbf{u}}^f$, of each surrogate function $\tilde{g}(\mathbf{u})$ and $\tilde{f}(\mathbf{u})$, respectively.

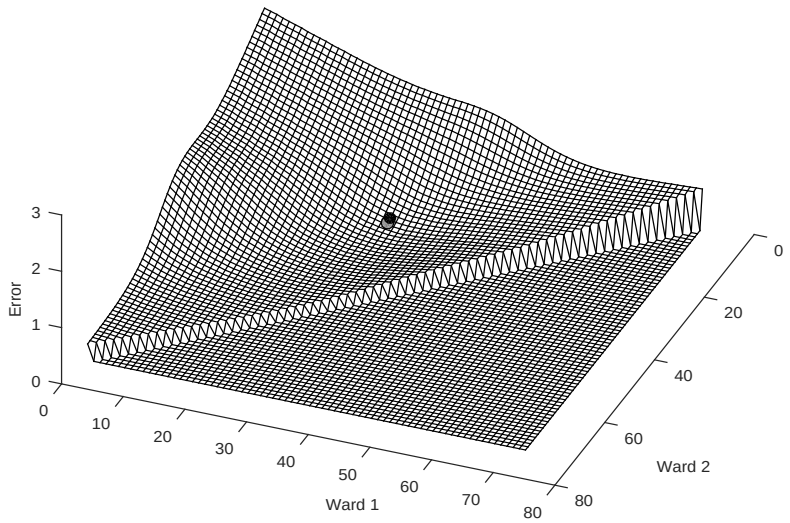


Figure 4.3: The error of surrogate objective function $\tilde{g}(\mathbf{u})$ on the original dataset, defined as $g(\mathbf{u}) - \tilde{g}(\mathbf{u})$, and derived by enumerating all solutions; showing, additionally, the optimum of $g(\mathbf{u})$ (black) and $\tilde{g}(\mathbf{u})$ (gray).

4.4.3 Evaluating the RIS Heuristic Parameters

The implications of adjusting the essential parameters are now explored. That is, the number of initial surrogate samples from the search space, the size of the exponent β, and the proximity tolerance ξ. All these parameters have been tested sequentially on the original data.

The results from adjusting the initial surrogate sampling is presented in Figure 4.4, showing the resulting interpolated estimate, $Y_g(\mathbf{u})$, and the associated runtimes based on 5, 20, 35 and 50 samples from the search space. Surrogate sampling is uniformly distributed and it is therefore expected that $Y_g(\mathbf{u})$ converges to the true function $g(\mathbf{u})$ when the sample size increases. Interestingly, the general shape of $g(\mathbf{u})$ can be determined fairly early, as shown in the experiment with only 5 samples.

By considering the strategic application of the RIS heuristic, we deem that the associated runtimes are fairly negligible, and since the apparent optimum does not change substantially after obtaining more than 20 samples, we deem that this is an adequate number of samples for our later optimization experiments.

Exponent β was assessed based on values of 1, 8 and 16. In accordance with the RIS heuristic, these experiments were conducted by first obtaining 20 uniformly distributed samples, followed by 20 samples according to the recursively-updated density function $Z(\mathbf{u})$. The surrogate objective function was again employed to conserve the runtime of the experiments. Results are presented in Figure 4.5, showing that the search intensifies around the apparent optimum as a function of β. Notice that the experiment where $\beta = 1$, corresponding to a complete omission of the conversion, demonstrates the usefulness of this approach as the sampling is almost uniformly distributed. In the experiment where $\beta = 8$, samples are generally close to the optimum, whereas in the last experiment, where $\beta = 16$, samples are concentrated on the apparent optimum with only a few outliers. Based on these experiments, a value of $\beta = 8$ is employed to focus on the most promising region of the search space, but still attains some diversification.

Lastly, the effect of adjusting the proximity tolerance, ξ, was assessed using values of 1, 4 and 8. The experiments were conducted by applying the full RIS heuristic using 20 initial surrogate samples, an exponent of $\beta = 8$, and an upper bound on the permitted number of relocated patients of $\tau = 1.9$. Each experiment was conducted using a runtime of 30 minutes.

Results of these experiments yielded almost identical performance in each case. This might have been caused by the choice of τ, which results in a rather limited sample space. Consequently, the proximity tolerance is rather arbitrarily set to $\xi = 4$ during the subsequent optimization experiments.

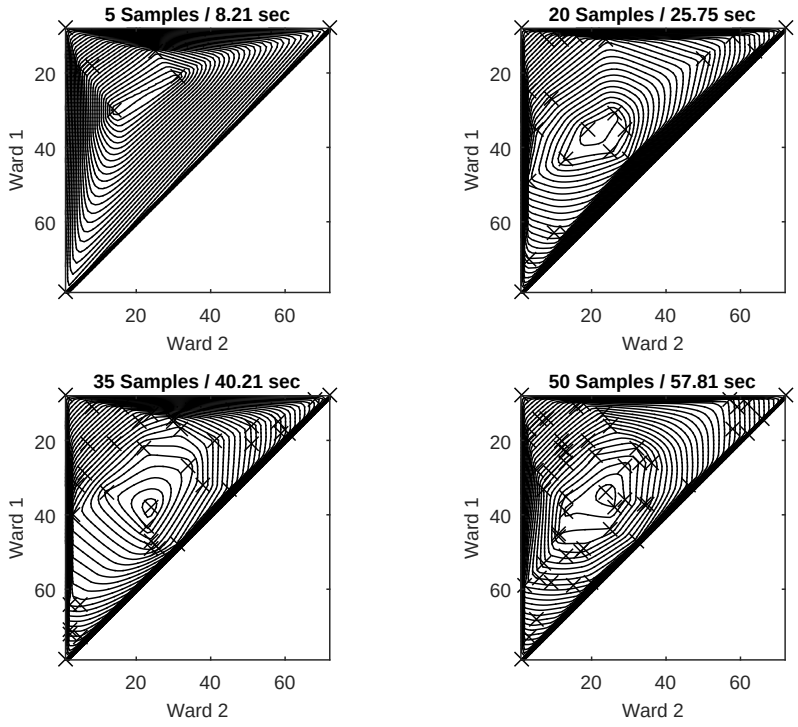


Figure 4.4: The result of gradually increasing the number of surrogate samples on the interpolated estimate $Y_g(\mathbf{u})$. $Y_g(\mathbf{u})$ does not change substantially upon applying more than 20 samples.

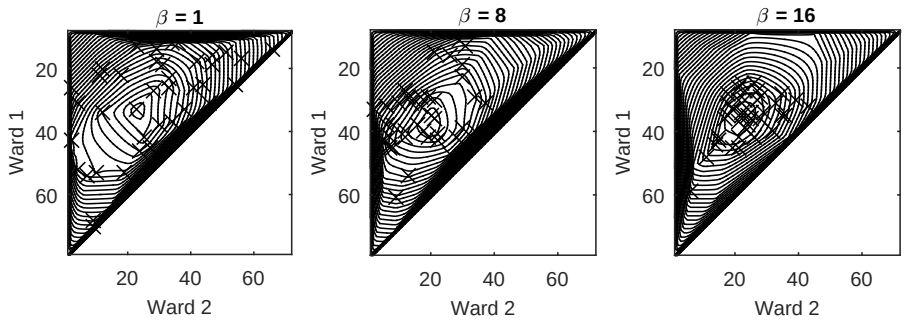


Figure 4.5: Interpolation $Y_g(\mathbf{u})$ resulting from a gradually increasing exponent, β. Each experiment is conducted by initializing with 20 uniform surrogate samples, followed by 20 samples based on $Z(\mathbf{u})$ (crosses).

4.4.4 Applying the RIS Heuristic

The full RIS heuristic is applied to the data presented in Section 4.4.1 based on the tests from Section 4.4.3. We begin by presenting an example of a single heuristic run, where gradually-obtained solutions are compared against the true optimum.

Overall performance is assessed by way of a number of experiments which compare the heuristic's solutions to the true optimum. Since no data was obtained concerning preference for private rooms, proportion ψ is investigated using three different levels. Furthermore, the robustness to changes in the patient arrival rates and relocation probabilities is of interest. Experiments will therefore be conducted on all three datasets (cf. Table 4.1 to 4.3).

A Single Run

Figure 4.6 illustrates the progression of the RIS heuristic on the *original* dataset for a runtime of 60 minutes and a bound of $\tau = \min\{f(\mathbf{u})\} \cdot 1.20 = 1.91$. During this time 21 iterations were conducted. The figure shows the interpolation $Y_g(\mathbf{u})$, samples $\mathbf{x}$ and $\hat{\mathbf{x}}$, the optimum obtained by enumeration, and finally the *estimated* and *true* feasible search space defined by τ.

The heuristic initializes with 20 surrogate samples, as shown in the upper left corner. At this stage, the apparent optimum is already close to the true optimum, which is immediately included within the search space. The remaining three graphs show the sampled solutions for iteration 5, 10 and 20. Notice that the estimated search space initially violates the true search space (in Figure 4.6, iteration 1 and 5: bottom-left solid line - estimated search space - exceeds dotted line - true search space), but then converges to the true search space near the optimum. At iteration 20 the estimated search space attains high accuracy near the true optimum, which decreases as more capacity is allocated to wards 1 and 2. Notice that samples are almost uniformly distributed due to the low slope near the optimum.

This example demonstrates the advantage of sampling from an interpolation based on a mix of both fast surrogate and slower true evaluations to determine the most promising region for an objective function of complex structure. The general performance of this approach for different parameter variations is investigated in the following section.

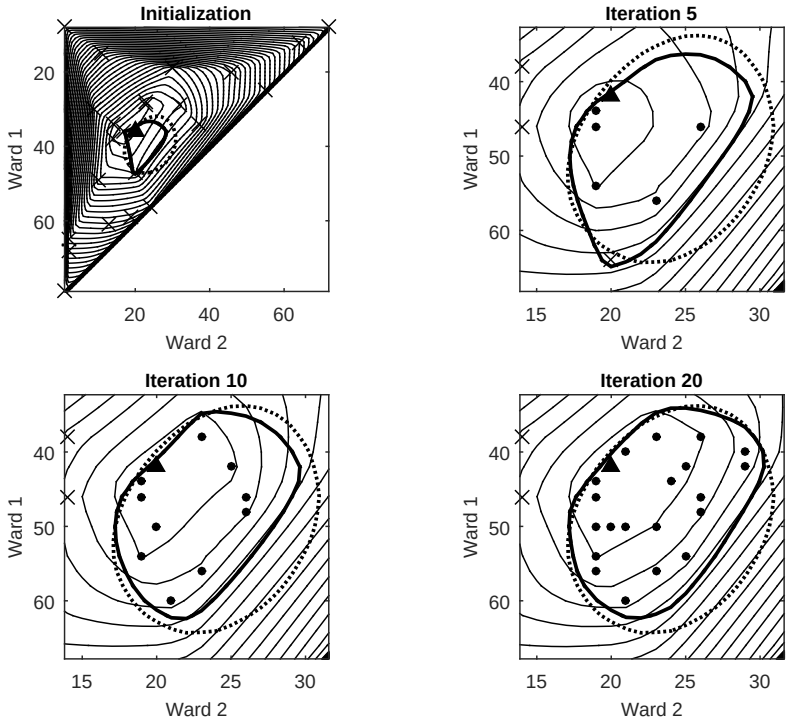


Figure 4.6: Progression of the RIS heuristic. Shows the interpolation, $Y_g(\mathbf{u})$, samples $\mathbf{x}$ (dots) and $\tilde{\mathbf{x}}$ (crosses) and the optimum (triangle). The true and estimated feasible search space is depicted with dotted and solid line, respectively.

Overall Performance

The overall RIS heuristic performance was assessed in two parts. Firstly, two runs were conducted on the *original* dataset for each of three levels of the private patient poportion $\psi = 0.2$, 0.5 and 0.7, with a fixed runtime of 60 minutes. To properly assess the heuristic solutions, results were compared against the true optima obtained by enumerating the search space.

Furthermore, since robustness regarding changes in the patient characteristics is of particular interest, similar runs were conducted for the *high arrival rate* and *high relocation* datasets. However, in order to determine the optima for these additional tests, the room availability was limited to $N_{private} = 20$ and $N_{double} = 10$ rooms. Due to the reduction of feasible ward capacity configurations, the relocation bound to yield the maximal search space was omitted,

and the runtime decreased to 10 minutes.

The results for the *original* dataset are presented in Table 4.5, featuring the heuristic room configuration, iterations corresponding to the number of solution-evaluations conducted using the true objective function, the heuristic objective value, the objective value for the optimum, and finally the gap between the best obtained solution and the optimum.

As expected, capacity is distributed among the wards according to the arrival rate of the three patient types. That is, with respect to both private and shared double rooms. By contrast, the fraction of private patients arriving, ψ, appears to have little effect on the distribution of private rooms, since the solutions are similar across all runs. This is potentially a result of the relative difference between the arrival rates of each patient type, which shall be assessed in the last part of this section, where the arrival rate has been increased for ward 1.

In general, the experiments presented in Table 4.5 yield excellent results, as the relative gap between the heuristic and true optimum is consistently below 1%. The reader should notice that these results have been obtained after a runtime of 1 hour, whereas the complete enumeration of the search space to determine the true optimum finished only after approximately 3 weeks runtime.

		Heuristic								**Optimal**	
ψ	Rep.	$u_{1,pr.}$	$u_{2,pr.}$	$u_{3,pr.}$	$u_{1,do.}$	$u_{2,do.}$	$u_{3,do.}$	Iter.	Obj. val.	Obj. val.	Gap (%)
0.2	1	13	11	12	8	6	5	28	12.68	12.75	0.55
0.2	2	15	10	11	8	5	6	20	12.71	12.75	0.31
0.5	1	15	11	10	8	5	6	23	30.09	30.28	0.63
0.5	2	15	10	11	7	6	6	29	30.18	30.28	0.33
0.7	1	15	10	11	7	6	6	20	35.21	35.23	0.06
0.7	2	14	10	12	8	5	6	27	35.20	35.23	0.09

Table 4.5: Result of optimizing the room configuration by applying our RIS heuristic. Each run has been replicated twice employing three different levels of the private patient fraction, ψ. The percent-wise difference from the optimum is shown in the last column.

Table 4.6 provides further results, showing the heuristic solutions for the *high arrival rate* and *high relocation* datasets. For the high arrival rate, the system is found to be more sensitive to changes in the private patient fraction since increasing the fraction results in more private rooms to be allocated to ward 1. Otherwise, the obtained solutions attain a small relative gap that consistently stays below 1%, showing that increasing the number of arriving patients does not affect the search procedure's performance.

Next, for a substantially larger number of relocated patients it is expected that more iterations are required to adapt the interpolation and obtain useful solutions, due to the lower accuracy of the surrogate objective function. Recall how it was previously determined that a substantial distance is present between the true and surrogate optima for the *high relocation* dataset in Section 4.4.2. The experiments indicate that this is only the case for a medium

			Heuristic								Optimal	
Dataset	ψ	Rep.	$u_{1,pr.}$	$u_{2,pr.}$	$u_{3,pr.}$	$u_{1,do.}$	$u_{2,do.}$	$u_{3,do.}$	Iter.	Obj. val.	Obj. val.	Gap (%)
High Arr.	0.2	1	9	5	6	7	2	1	53	7.57	7.62	0.66
High Arr.	0.2	2	9	5	6	7	2	1	64	7.57	7.62	0.66
High Arr.	0.5	1	9	2	9	5	1	4	74	17.35	17.45	0.57
High Arr.	0.5	2	11	2	7	5	1	4	78	17.36	17.45	0.52
High Arr.	0.7	1	14	1	5	6	1	3	70	19.76	19.77	0.05
High Arr.	0.7	2	12	0	8	5	1	4	83	19.76	19.77	0.05
High Rel.	0.2	1	5	4	11	3	0	7	44	7.70	7.73	0.39
High Rel.	0.2	2	5	3	12	1	1	8	18	7.72	7.73	0.13
High Rel.	0.5	1	2	8	10	1	4	5	3	17.30	17.71	2.32
High Rel.	0.5	2	8	1	1	4	1	5	34	17.45	17.71	1.47
High Rel.	0.7	1	6	0	14	3	1	6	11	19.82	19.85	0.15
High Rel.	0.7	2	1	2	17	2	2	6	2	19.76	19.85	0.45

Table 4.6: Results of applying our RIS heuristic to the *high arrival rate* and *high relocation* datasets. The availability of rooms were limited to $N_{private} = 20$ and $N_{double} = 10$. Due to the limited search space, all runs were conducted without the relocation bound and a runtime of 10 minutes.

private patient fraction of $\psi = 0.5$ since the relative gap has increased a few percentage points. The other levels remain relatively unchanged, and even slightly improved for $\psi = 0.2$. Thus, these experiments indicate that good solutions are derived for a large number of relocated patients as well.

4.4.5 Validation

Finally, this section validates the assumption that an improved room configuration leads to better operational efficiency for inpatient admissions. An initial, poor quality, room configuration (RC1) is compared with an optimized room configuration (RC2, determined as a result of the RIS heuristic) in a day-to-day scheduling simulation. The simulation begins from an initial, empty set of wards with given room configuration (either RC1 or RC2). As the simulation time progresses, inpatients arrive according to the arrival rates determined by Table 4.1. At the start of each day a simplified version of the reactive ILP model [131] is solved. This results in the patient-to-room and -ward assignments for the considered day, and patients will stay in their assigned room until they are discharged.

The ILP model can be defined as follows. Let binary decision variable x_{pr} equal 1 if patient p (which either arrived on the current day, or is still present from previous admission) is admitted to room r. Let binary decision variable y_{rt} equal 1 if on day t the room is assigned to male patients or 0 if assigned to female patients. Finally let $r = \emptyset$ denote a *dummy* room, where $x_{p\emptyset} = 1$ indicates a patient being refused (or being relocated to a ward which is not considered in the current problem). Also note that this section redefines variables/indices used in the previous sections.

Then the ILP optimization model can be formulated as:

$$\text{Minimize} \sum_{p \in P} \sum_{r \in R} c_{pr} \cdot elos(p) \cdot x_{pr} \tag{4.24}$$

$$\text{Subject to} \tag{4.25}$$

$$\sum_{r \in R} x_{pr} + x_{p\emptyset} = 1 \qquad \forall p \in P \tag{4.26}$$

$$\sum_{\substack{p \in P: \\ t < ad(p) + elos(p)}} x_{pr} \leq b_r \qquad \forall r \in R, t = d, \ldots, D \tag{4.27}$$

$$\sum_{\substack{p \in P: \\ t < ad(p) + elos(p) \\ p = \text{male}}} x_{pr} \leq b_r \cdot y_{rt} \qquad \forall r \in R, t = d, \ldots, D \tag{4.28}$$

$$\sum_{\substack{p \in P: \\ t < ad(p) + elos(p) \\ p = \text{female}}} x_{pr} \leq b_r \cdot (1 - y_{rt}) \qquad \forall r \in R, t = d, \ldots, D \tag{4.29}$$

$$x_{pr} \in \{0, 1\} \qquad \forall p \in P, r \in R \cup \emptyset \tag{4.30}$$

$$y_{rt} \in \{0, 1\} \qquad \forall r \in R, t = d, \ldots, D \tag{4.31}$$

with R denoting the set of rooms available from the wards in $\mathcal{W}$. b_r denotes the capacity of each room $r \in R$, i.e. in the current dataset $b_r \in \{1, 2\}$ (private or shared). P denotes the set of patients currently arriving for admission or still present after admission on an earlier day in the simulation. $ad(p)$ and $elos(p)$ denote the arrival day and *expected length of stay* (available from Table 4.1, $1/\mu$ parameter for each patient type) in days. D denotes an upper bound on the planning horizon (in days), which is restricted by either the length of the simulation time horizon or the maximum remaining expected length of stay among patients $p \in P$. Finally, c_{pr} denotes a cost matrix, attributing a perceived penalty of admitting a patient p to a room r for one day. The elements c_{pr}, defined for each room (including the dummy) and patient combination, are given by the sum of:

- w_{pref}, a room preference penalty if the assigned room does not meet the patients preference (i.e. a shared room when the patient prefers private),
- $(1 - p_{ij}) \cdot w_{\text{reloc}}$, a relocation penalty if the assigned room is not in the preferred ward (refer to Table 4.2 for values of p_{ij}),
- $w_{\emptyset}$, a refusal penalty if the patient is not admitted to a room from wards $\mathcal{W}$.

The simulation was run on the three datasets described in Section 4.4.1[4]. Room configurations RC1 and RC2 depend on the dataset, and are constructed as described in Table 4.7. Furthermore, the fraction of male patients

[4]The above formulation does not include the constraint that prevents previously admitted patients from being re-allocated. However, this constraint is included in the simulation experiments.

Dataset	Room config.	Ward 1		Ward 2		Ward 3		Obj. val
		# Private	# Shared	# Private	# Shared	# Private	# Shared	
Original	RC1	5	5	25	4	6	10	26.1
Original	RC2	15	7	10	3	11	6	35.2
High Arr.	RC1	5	5	25	4	6	10	26.3
High Arr.	RC2	18	8	9	6	9	5	35.5
High Rel.	RC1	5	5	25	4	6	10	26.0
High Rel.	RC2	14	8	11	6	11	5	35.2

Table 4.7: Room configurations for each dataset applied in the simulation.

Dataset	ϕ	Avg. diff. room pref.	Avg. diff. reloc.	Avg. diff. refusal	Avg. diff	p-value
Original	0.3	339.672	1714.919	-0.156	1524482.324	**0.000003**
Original	0.4	335.76	1714.122	0.342	1573963.133	**0.000003**
Original	0.45	333.636	1716.732	-1.312	1410879.453	**0.000011**
Original	0.5	330.499	1714.839	-0.212	1519234.307	**0.000003**
Original	0.55	328.462	1714.334	-0.05	1535030.136	**0.000002**
Original	0.6	330.563	1713.348	-0.117	1527145.516	**0.000003**
Original	0.7	335.4	1715.605	-0.794	1461706.145	**0.000009**
High Arr.	0.3	209.789	2327.362	-0.655	1978448.163	**0.004524**
High Arr.	0.4	209.705	2328.767	-0.753	1969645.429	**0.004796**
High Arr.	0.45	209.596	2324.194	2.314	2272155.253	**0.000699**
High Arr.	0.5	208.725	2331.665	-3.023	1745265.698	**0.013875**
High Arr.	0.55	209.611	2327.468	-0.056	2038268.366	**0.003074**
High Arr.	0.6	210.234	2327.527	-1.443	1899492.767	**0.007611**
High Arr.	0.7	211.017	2326.236	-0.516	1991301.962	**0.004637**
High Rel.	0.3	313.305	1835.305	-0.142	1066569.388	**0.001142**
High Rel.	0.4	308.111	1837.001	-1.632	918296.2456	**0.003782**
High Rel.	0.45	307.47	1835.729	-1.371	943633.4047	**0.002924**
High Rel.	0.5	309.619	1833.69	-0.969	983328.3526	**0.003197**
High Rel.	0.55	307.042	1834.621	-0.642	1016144.088	**0.001414**
High Rel.	0.6	311.93	1834.392	-1.098	970200.3081	**0.003658**
High Rel.	0.7	312.194	1836.225	-1.501	931214.4434	**0.003905**

Table 4.8: Simulation results (averaged over 1000 replications) for each dataset and male patient fraction.

among all patients (other patients being female) is considered as an additional parameter (denoted by ϕ) since this fraction may be ward-dependent. The simulation runs over 120 simulation days during which arrivals are generated. For each combination of the considered parameters (fraction of male patients, dataset and room configuration), we ran 1000 simulation replications.

The results (averaged over 1000 replications) are summarized in Table 4.8, showing for each dataset and male patient-fraction, the difference between RC1 and RC2 (value larger than 0 if RC2 is better) in respectively room preference penalties, patient relocations to other wards, patient refusals, and the global objective value as defined by Equation (4.24) calculated over the simulation horizon of 120 simulation days. Finally, the difference is statistically significant by a p-value from the Wilcoxon Rank Sum Test. Notice that all p-values are indeed smaller than 0.05.

These results show that RC2, while having a *marginally* higher number of patient refusals, the penalties of mismatched room preferences (our primary concern) and patient relocations greatly improves over RC1. This validates the assumption that an improved room configuration (i.e. RC2 over RC1) leads to improved operational efficiency for inpatient admissions.

4.5 Conclusion

The ability to choose a private room over a shared room is becoming an increasingly important factor for patients to choose a hospital for admission. Being able to meet those requests is of strategic importance to hospitals, both in increasing patient comfort and satisfaction but also in generating extra revenue from charging room/honorarium supplements. However, existing infrastructure may not be adequately allocated between nursing wards to meet the current demand.

This study sought to provide hospital decision makers with a strategic tool for improving the allocation of room types among hospital wards. More specifically, the aim was to accommodate patients who prefer private room assignments, by first assuming a fixed number of room types, and second that these room types can be reallocated among the wards.

The proposed approach is based on a continuous-time Markov chain model that derives the patient occupancy distributions, and a heuristic search procedure referred to as Randomized and Interpolated Search (RIS) that searches for the best possible room configuration. RIS recursively improves an initial solution by sampling from the search space based on a gradually improved interpolation of the objective function. The fact that occupancy distributions are fixed for an unchanged aggregated capacity is exploited in order to derive the sub-optimal room type configuration using integer linear programming. Consequently, aggregated capacity allocations form the primary decision variables for the proposed RIS heuristic to operate on. This results in reducing the search space for the RIS heuristic by omitting room type configuration decision variables.

Based on data from both a Danish and Belgian hospital, the applicability and effectiveness of the approach was demonstrated through various experiments which vary the fraction of patients who prefer private hospitalization, the overall arrival rate, and lastly the number of patients relocated within the system. In a computational study, it is shown that the RIS heuristic has the potential to derive near-optimal solutions that attain relative gaps below 1% within short runtimes which make the method applicable in practice. Moreover, it was demonstrated how configuring room resources on a strategic level benefits the day-to-day decisions of assigning patients to rooms through simulation.

Finally, the reader should notice that the proposed approach is not only applicable to the specific case of optimizing the room configurations in a hospital setup, but to any queueing problem where jobs are serviced among different nodes and may prefer a specific, but limited resource. Examples of such similar environments vary from manufacturing setups where products are processed at different stations and may require a specific tool, to call centers where customers may prefer an operator of a specific skill set.

The study comprised various experiments, including different parameter

variations as well as input datasets that demonstrate the performance and the robustness of the approach. However the analysis was restricted to a specific hospital case featuring a fairly limited number of wards. More complex cases should be assessed, preferably with a greater number of disposable rooms and room types. Additionally, future experiments should consider that patient scheduling is not only constrained by room preferences but also gender, and that this mix can be a function of the preferred ward for the patient.

Acknowledgments

Wim Vancroonenburg is a post-doctoral researcher funded by Research Foundation Flanders - FWO Vlaanderen. This research was further funded and supported by the Danish governmental organization, Region Sjælland. Especially, we thank the department of Production, Research and Innovation for providing us with insight into the operations of the Danish hospitals. Lastly, we thank Prof. Bo Friis Nielsen at the Technical University of Denmark for providing us with advice on probability theory. Editorial consultation provided by Luke Connolly (KU Leuven).

Part III

Acute and Surgical Flow

Chapter 5

Staff Optimization for Time-Dependent Acute Patient Flow[1]

Anders Reenberg Andersen, Bo Friis Nielsen,
Line Blander Reinhardt and Thomas Jacob Riis Stidsen

Abstract The emergency department is a key element of acute patient flow, but due to high demand and an alternating rate of arriving patients, the department is often challenged by insufficient capacity. Proper allocation of resources to match demand is, therefore, a vital task for many emergency departments.

Constrained by targets on patient waiting time, we consider the problem of minimizing the total amount of staff-resources allocated to an emergency department. We test a matheuristic approach to this problem, accounting for both patient flow and staff scheduling restrictions. Using a continuous-time Markov chain, patient flow is modeled as a time-dependent queueing network where inhomogeneous behavior is evaluated using the uniformization method. Based on this modeling approach, we recursively evaluate and allocate staff to the system using integer linear programming until the waiting time targets are respected in all queues of the network. By comparing to discrete-event simulations of the associated system, we show that this approach is adequate for both modeling and optimizing the patient flow. In addition, we demonstrate robustness to the service time distribution and the associated system with multiple classes of patients.

5.1 Introduction

In this study, we consider the well-known problem of optimizing the patient flow for an Emergency Department (ED). With many hospitalizations on a daily basis, the ED is often considered a vital element to the hospital compared to other hospital departments. ED hospitalizations are further characterized by a large variety of different diagnoses, requiring staff from a range of different specializations around the clock.

A report from the Danish Ministry of Health [97] places Denmark below the average lifespan for countries in the Organization for Economic Co-operation

[1] Accepted for publication in the European Journal of Operational Research

and Development (OECD), but above the average on fraction of Gross Domestic Product (GDP) used on public health care; hence suggesting that a general increase in the utilization of resources is required. In this study, we address this issue by providing hospital management with a method for deriving the minimum required staff for an ED constrained by targets on patient waiting time. Such method is especially relevant for hospitals that are governed by their efficiency, and therefore seek to rearrange the excess resources for instance by validating the difference between the minimum required and currently available resources.

Operations Research literature related to Emergency Department (ED) planning and dimensioning is relatively unexplored as regards analytical modeling of acute patient flow combined with optimization.

Lim et al., 2012 [85] conducted an elaborate survey on the use of mathematical modeling of ED patient waiting times and found 29 relevant studies. From these, four overall modeling techniques were uncovered: (1) Queueing Theoretic (QT) models covered a total of four different studies, (2) Discrete Event Simulation (DES) covered 22 different studies, (3) System Dynamics (SD) covered two studies and (4) Agent-Based Modeling (ABM) covered two studies likewise. Substantial weight is obviously given to the three simulation-related approaches as only four studies were conducted using QT modeling.

Lim et al., 2012 further found that a recurrent objective is to use the model to test one or more scenarios and rarely to optimize the system. Examples in QT modeling are Cochran & Roche, 2009 [42] and Mayhew & Smith, 2008 [88], where open queueing network models are developed with a view to investigate how to increase patient throughput. In the area of DES, Medeiros et al., 2008 [91] tested an approach named Provider Directed Queueing for improving ED performance. Additionally, Khadem et al., 2008 [76] assessed a new layout for an ED and found the new layout to reduce patient waiting time by a substantial amount. In SD modelling, Storrow et al., 2008 [124] assessed the effect of decreasing lab turnaround times, focusing on emergency medical services, patient throughput and length of stay. Lane et al., 2000 [81] assessed changes in waiting times as bed capacity is changed. Further, in the area of ABM, Wang, 2009 [136] evaluated different settings of triage and radiology procedures. Lastly, some studies combine different modeling approaches to attain their objective. Laskowski et al., 2009 [82] evaluated patient flow using two different models. One based on ABM and the second based on queueing theory. In their study, the two models are applied and compared by using a number of relatively simple scenarios.

Getting an understanding of acute patient flow based on simulation seems well explored. However, Lim et al., 2012 only obtained two studies that use modeling of patient flow in an actual optimization scheme. The first is Yeh & Lin, 2007 [143] where schedules are adjusted for a fixed amount of nurses by using a combination of DES and a Genetic Algorithm (GA). The aim was to find the configuration of schedules that minimizes patient waiting time. Secondly, Ahmed & Alkhamis, 2009 [10] combined DES with a local search heuristic by

applying statistical hypothesis testing. The goal was to determine the optimal number of different staff types by maximizing the throughput of patients constrained by department budgets.

Besides the studies in Lim et al., 2012 we were able to identify four studies where optimization is conducted in the context of acute patient flow. Firstly, Sinreich et al., 2012 [119] use a DES model together with Mixed Integer Linear Programming (MILP) to derive two different heuristics with the aim of determining efficient work-shift schedules that minimize patient waiting time. Further, Daldoul et al., 2015 [44] determined the optimal amount of staff and equipment by using a MILP model. Interestingly, system stochasticity was not incorporated in this model. In addition, Cabrera et al., 2012 [30] used ABM and exhaustive search to optimize the configuration of different staff types, and lastly, Wang, 2013 [137] used a modeling approach known as Separated Continuous Linear Programming to determine the level of staffing that would minimize the overall cost of the ED.
Now, when we consider studies that focus only on queueing theoretic modeling, then queues with non-homogeneous Poisson arrivals or even processes with more general time dependent arrivals has received substantial interest. See e.g. Schwarz et al., 2016 [113] and Defreye & Inneke, 2016 [46] for two recent review papers. The literature on time dependent queuing networks specifically is less abundant, but see Armony et al., 2015 [15] for a data-based analysis of ED's viewed through the lens of a queueing scientist.

Moving to different application areas, in manufacturing Bitran & Morabito, 1994 [24] conducted a survey on stationary open queueing networks, presenting both exact and approximate solutions to a range of different problem structures. Related to our study, the problem of minimizing cost by allocating machines, constrained by an upper bound on a Work-In-Progress (WIP) level, may be solved approximately by a heuristic. However, if the number of machines is fixed, and the objective is to minimize the WIP level, then an exact solution can be derived.

Further, on optimizing stationary queueing networks, Smith et al., 2010 [120] present an exact solution to the machine allocation problem for a finite queueing network by using Powell's algorithm. For a general open queueing network, Giloni, 2001 [56] derives conditions under which the problem is reduced to solving a concave or convex problem. Additionally, Seshadri & Pinedo, 1999 [114] exploit an approach where a heuristic is used to minimize the WIP level. Lastly, Yoneda et al., 1992 [144] apply simulated annealing to optimize their system.

In the area of call center staffing, several studies have been conducted considering both time-varying arrivals and staff optimization. For single queues, Feldman et al., 2004 [53] investigate three different methods for deriving the minimal time-dependent staffing level, s_t, to maintain time-stable performance. The study proposes a simulation-based algorithm, along with an extension of the square-root-staffing formula [73]. Lastly, for queues with customer aban-

donments, $M_t/M/s_t + M$, they show for a certain setting that staffing can be adjusted to match the expected load in the associated infinite-server system. Related hereto, Whitt, 2006 [138] maximizes the revenue of an $M/GI/s + GI$ queue by firstly modeling the system as a deterministic fluid model, and secondly as the associated $M/M/s+M(n)$ model. The optimization is conducted by adjusting the number of servers in the system. Further, Sze, 1984 [125] focuses on choosing an adequate $M/G/s$ model for staffing purposes, taking arrival variability into account.

Turning to queueing networks in call center staffing, Tipper & Sundareshan, 1990, [127] consider a network of single-server queues with time-varying arrival rate, using two models. The first is based on Chapman-Kolmogorov differential equations, and the next on non-linear differential equations modeling the mean queue lengths in the network. We have noticed that neither in this study nor the proceeding three studies on single queues is emphasis put on incorporating staff in more complex shift structures. Liao et al., 2012 [84] derives the optimal staffing level of a single queue, incorporating back-office jobs, by using both stochastic and robust programming, respectively, but assumes a single-shift structure.

We acknowledge that modeling of queueing networks is an extensive field covering many other applications as have not been mentioned above. In this review, we mainly focus on the literature covering optimization of acute patient flow, and two related application areas. In the area of manufacturing, non-stationary cases seem to be rarely considered, whereas for call center modeling and staffing, limited emphasis is put on optimizing staff with more elaborate shift structures. In the area of modeling flow for acute patients queueing theory combined with optimization is in general an uncommon approach.

In our study, we present an approach based on a continuous-time Markov chain (CTMC) for modeling the time-dependent behavior of acute patient waiting time, and the interaction of this approach with an Integer Linear Programming (ILP) model. The ILP will serve as the method we use to efficiently allocate staff to specific working-patterns, as has been proven adequate by other studies [39]. We combine the CTMC and ILP in a matheuristic search procedure with the objective of minimizing the total amount of staff that is allocated to the ED. This heuristic procedure is further divided into two variations, yielding two models for our numeric experiments. Further, we have used a Danish ED as the basis for our study and have constructed a representation of patient flow as well as conducted tests based on data from this ED.

Specifically, our contribution to the area of acute patient modeling and optimization is:

- Applying an analytical approach for modeling time-dependent flow for acute patients, going from triage to specialized treatment. Specifically, we employ a numerical method for modeling the system as an open queueing network.

- Combining the queueing network with an ILP model in a simple and generalizable matheuristic procedure for minimizing staff, taking constraints on patient waiting time, as well as staff working-patterns into account.

In Section 5.2 we elaborate on the specific problem and data at hand. In Section 5.3, we present the CTMC model that is used to evaluate patient waiting time and the structure of the matheuristic incorporating both CTMC and ILP modeling. In Section 5.4 we evaluate our CTMC approach, present how the tuning of parameters is conducted, and demonstrate the performance of our matheuristic. Lastly, we present our conclusion in Section 5.5.

5.2 Problem Description

Any patient who is admitted to an Emergency Department (ED) will be dependent on a range of different resources. Upon arrival, the patient is firstly triaged to determine the severity of the patient's condition. Next, an examination is conducted by a physician to determine whether a more in-depth treatment is required. If there is no need for this, the patient can be discharged immediately; otherwise, a specialized physician is further required.

Obviously, the admission of a patient involves the use of a range of many different resources. In case one or more of these resources are absent, the treatment quality will decrease accordingly. Still, like any other organization the ED is subject to a limited capacity and is thus faced with the problem of balancing quality of care against the department expenditures. Being able to make clear objectives and utilize resources accordingly is, therefore, a core responsibility of the department.

Our objective is to contribute to the methodology related to balancing ED capacity against service, by minimizing the total amount of staff allocated to the department taking the resulting effect on treatment quality into account. As waiting time has been shown to directly influence the treatment quality of acute patients [99, 72, 93], the total amount of allocated staff will be constrained by targets on patient waiting time. Due to union settlements, we further consider that staff resources are constrained by a number of fixed working patterns.

5.2.1 System and Data Description

We consider patients of a single class arriving according to a process with time-varying intensity to an ED. Upon arrival, the patients are physically admitted to a bed, where they will stay until discharged. During this time, however, the patients require attention from a range of different staff types depending on their diagnosis. Each staff type is drawn from a "pool" of limited capacity and will attend the patients for a random amount of time. Thus, we assume that the stay of a patient can be modeled as an open queueing network, where the change in required attention between staff types corresponds to moving

from one network node to the next. Due to this approach, we assume that a patient can only be treated by a single care-provider at a time.

In case a patient requires attention from a pool of staff where all members are occupied, a queue is created, and the patient will receive attention according to a *first-come first-served* (FCFS) discipline. To represent the diversity of diagnoses and need, the routing of patients from one queue to the next occurs randomly, but with known probability. Additionally, in case a patient requires attention from the same care-provider more than once, the patient is looped back to the same node. Specifically, we interpret the patient flow as the queueing network presented in Figure 5.1. Here, each queue of the network represents the following five staff types (by queue number):

1. Triage Nurses
2. Basic Physicians
3. Specialized Medical Physicians
4. Organ Surgeons
5. Orthopedic Surgeons

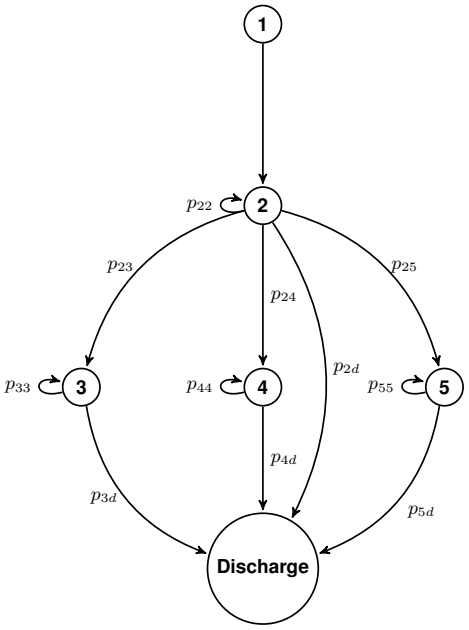


Figure 5.1: The stay of patients modeled as a network of queues. Each node represents a single queue with servers of only one staff type. Staff types by queue number are: (1) Triage Nurses, (2) Basic Physicians, (3) Specialized Medical Physicians, (4) Organ Surgeons and (5) Orthopedic Surgeons. The parameter, p_{ij}, defines the routing probability.

Patient Data

We obtained one year of patient data from a Danish ED, showing the exact arrival time and triage level of each patient. Our case-ED uses four triage levels between which patients are initially distributed (in ascending priority) with 9% on level 1, 63% on level 2, 25% on level 3, and 3% on level 4. Furthermore, we naturally found that all patients were triaged on arrival, but then have their priority level adjusted after the first examination by a physician. That is, after the examination about 72% of the patients on level 3 were re-evaluated to level 2, essentially changing the distribution to 81% of the patients on level 2 and only 7% on level 3 for the remaining queues in the network.

Based on the ED data, we further investigated the patient inter-arrival time by modeling the arrival rate as the Poisson regression, shown in (5.1),

$$log(\lambda_{ij}(u)) = \alpha+\beta u+\theta u^2+\gamma_j+\delta_i+\phi_j u+\zeta_j u^2+\psi_i u+\xi_i u^2+\rho_{ij}+\eta_{ij} u+\omega_{ij} u^2 \tag{5.1}$$

where $\lambda_{ij}(u)$ is the expected number of arrivals on hour of the day $\{u \in \mathbb{R}|0 \leq u \leq 24\}$, on the day of the week $j \in \{Monday, Tuesday, \ldots, Sunday\}$ for patients of triage priority $i \in \{1, 2, 3, 4\}$. We used explanatory variables to model the effect of day of the week, j, and triage priority i. Due to the limited amount of data obtained, we modeled the effect from time of the day, u, as a second order polynomial. The resulting modeled arrival rate is demonstrated in Figure 5.2, showing both the modeled and empirical rates for patients of triage level 2 and 3, respectively.

We evaluated our model by examining the distribution of $\epsilon = (y - \hat{\lambda})/\sqrt{\hat{\lambda}}$, where $\hat{\lambda}$ is the model fit and y the observations. In addition, we conducted a graphical test where the model was fitted to the first six months of data and then compared to the last six months. Lastly, we estimated the dispersion parameter at $\hat{\phi} = 0.82$, and conducted a Pearson's goodness-of-fit based on a model deviance of 41346 with 50274 degrees of freedom, yielding a right-tailed probability of $p = 1$. Thus, a very large p-value. From these both graphical and quantifiable measures we have found Poisson behavior to fit the data well.

As patients are admitted to the ED, they will require attention from a range of different staff types, which we interpret as the service times of the queueing network. For the case ED, we were not able to obtain reliable data on time spent on patients. Therefore, in our subsequent modeling we will assume for convenience that service times are exponentially distributed, even though we acknowledge that such distribution might not fit real-life inter-service times of an ED. Later, in Section 5.4.1, we elaborate more on the robustness of this assumption. The specific parameters that we have used for our service time distributions are presented in Appendix B.1, Table B.1. These were estimated based on interviews with hospital staff.

Lastly, regarding the use of prioritizes, recall that a fairly large fraction of

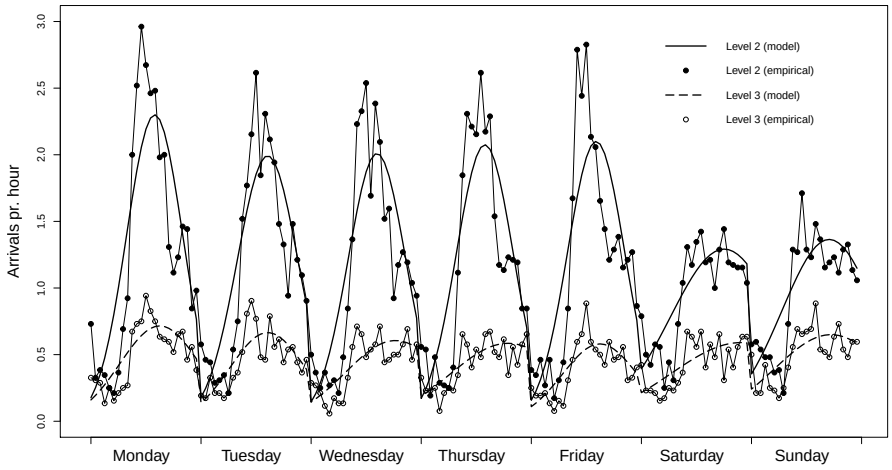


Figure 5.2: The modeled, according to (5.1), and empirical time-varying arrival rate for patients of triage level 2 and 3 respectively.

patients are prioritized on triage level 2, especially after the examination by a physician. Therefore, to ensure computational tractability of our modeling approach, we will only be considering a single class of patients with arrival rate corresponding to the sum of triage level 2 and 3. Since we only consider this single class, we may drop the index on triage level, such that the arrival rate simplifies to $\lambda_j(u)$. Later, in Section 5.3.1, we will refer to the arrival rate as $\lambda(\xi)$, where ξ is any continuous point in time within one week. We elaborate more on the implications of this assumption in Section 5.4.2.

Besides the arrival rates derived from (5.1), we test two additional arrival patterns for our optimization experiments, $g_j(u) = \lambda_j(u) \cdot 0.9$ and $h_j(u) = \lambda_j(u) \cdot 0.9^2$, depicted in Figure 5.3.

Routing

Depending on condition and diagnosis, any patient arriving at the ED will have a unique need for care, and as a result, there is a large range of different combinations of services to account for in the patient flow. From the perspective of our queueing network, the necessary care will be reflected in the patient being routed to either a specialized physician, looped to the current care provider, or discharged. The question is whether such routing occurs randomly or depends on some underlying policy. For instance, in situations of overcrowding at the optimal care resource, it might be worth to consider an alternative option for the patient. However, our interviews with the ED staff indicated that the patients are provided with the care optimal to each patient, in which case routing occurs randomly in accordance with the random occurrence of condition

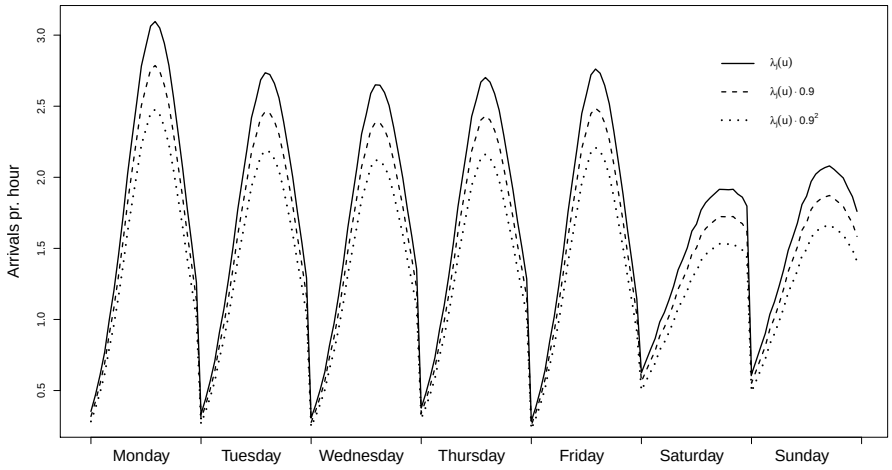


Figure 5.3: The three arrival rate patterns used to test the performance of our matheuristic approach.

and diagnosis. Furthermore, we found that patients require care resources according to some distribution. Let p_{ij} define the probability of being served by staff type $j \in C$ successive to $i \in C$, and let p_{id} define the probability of being discharged and leaving the system upon completion of $i \in C$, as shown in Figure 5.1. To derive the value of these, we obtained patient data showing the specific staff resources that were required during each patient's stay. The result is presented in Appendix B.1, Table B.2.

5.3 Modeling & Solution Approach

In this section we present an approach for the problem of minimizing the total amount of staff allocated to the ED, constrained by targets on the patient waiting time. We consider a set of staff types, C, subject to a limited set of working-patterns, J, and that patient waiting time is a non-linear function of the available capacity in the department. This leads to the master problem shown in equation (5.2a)-(5.2d),

$$\text{min.} \quad \sum_{c \in C} \sum_{j \in J} x_{cj} \tag{5.2a}$$

s.t.

$$L_{ct}(\mathbf{Z}) \geq \tau \qquad \forall t \in T,\, c \in C, \text{ where } z_{ct} = \sum_{j \in J} a_{cjt} x_{cj} \tag{5.2b}$$

$$\sum_{j \in J} a_{cjt} x_{cj} \geq \beta_{ct} \qquad \forall t \in T,\, c \in C \tag{5.2c}$$

$$x_{cj} \in \mathbb{N}_0 \qquad \forall j \in J,\, c \in C \tag{5.2d}$$

where x_{cj} is the amount of staff type $c \in C$ assigned to working-pattern $j \in J$. Thus, (5.2a) is the total amount of staff allocated to the department. Furthermore, $L_{ct}(\mathbf{Z})$ is the fraction of patients waiting for staff type $c \in C$ below a predefined time, in time period $t \in T$, where T is a discrete set of the weekly hours $T = \{1, 2, \dots, 168\}$. Here, $\mathbf{Z}$ is a $|T| \times |C|$ matrix defining the resulting allocation of each staff type for each time period in the entire planning period. Let z_{ct} be an element of $\mathbf{Z}$, and let $a_{cjt} \in \{0, 1\}$ be equal to 1 if working-pattern $j \in J$ assigns staff type $c \in C$ to time period $t \in T$; otherwise 0. Then, $z_{ct} = \sum_{j \in J} a_{cjt} x_{cj}$, is the amount of staff type $c \in C$ allocated to time period $t \in T$. Since $\{\tau \in \mathbb{R} | 0 < \tau < 1\}$, (5.2b) constraints the fraction of patients with a waiting time below a predefined amount of time.

Lastly, we assume that the system has a limit for each staff type $c \in C$ and time period $t \in T$, β_{ct}, after which the system is no longer operative. Constraints (5.2c) is introduced to ensure that the staff limit is never violated.

We evaluate $L_{ct}(\mathbf{Z})$ by using a continuous-time Markov chain, presented in Section 5.3.1. Due to the non-linear and complex structure of $L_{ct}(\mathbf{Z})$ there exists, to our knowledge, no standard approach to solve (5.2a)-(5.2d). We present a heuristic approach in Section 5.3.2.

5.3.1 Modeling Patient Waiting Time

As previously mentioned, we consider five staff types, $C = \{1, 2, \dots, 5\}$ interpreted as five different nodes in a queueing network. To model the occupancy and flow between these queues, we introduce a continuous-time Markov chain (CTMC) with state definition $s = \{k_1, k_2, \dots, k_5\}$, where k_i is the number of patients waiting for *or* in service by staff type $i \in C$. We further consider a truncation of the patient capacity, $M_i \geq k_i$, and choose this so the probability of having M_i patients waiting for or being served by staff $i \in C$ has negligible effect on the behavior of the system. Then the CTMC has state space $S = \{0, \dots, M_1\} \times \{0, \dots, M_2\} \times \cdots \times \{0, \dots, M_5\}$ of size $|S| = \prod_{i \in C}(M_i + 1)$.

Furthermore, let $\lambda(\xi)$ define the arrival rate of a single class of patients at time ξ. In addition, let μ_i define the service rate of staff type $i \in C$. Moreover, let w_i define the number of servers of staff type $i \in C$, and assume that $w_i < M_i$.

Let Q define the transition rate matrix of the CTMC, with q_{ss^*} the transition rate from the current state $s \in S$ to a new state $s^* \in S$. Then we have,

$$q_{ss^*} = \begin{cases} \lambda(\xi) & \text{if } s^* = (k_1 + 1, k_2, \dots, k_5) \text{ and } k_1 < M_1 \\ \mu_1 k_1 & \text{if } s^* = (k_1 - 1, k_2 + 1, \dots, k_5) \text{ and } k_1 > 0, k_2 < M_2, k_1 \leq w_1 \\ \mu_1 w_1 & \text{if } s^* = (k_1 - 1, k_2 + 1, \dots, k_5) \text{ and } k_1 > 0, k_2 < M_2, k_1 \geq w_1 \\ \mu_2 k_2 p_{2j} & \text{if } s^* = (k_1, k_2 - 1, \dots, k_j + 1, \dots) \text{ and } k_2 > 0, k_j < M_j, k_2 \leq w_2 \quad \forall j \in C \setminus \{1, 2\} \\ \mu_2 w_2 p_{2j} & \text{if } s^* = (k_1, k_2 - 1, \dots, k_j + 1, \dots) \text{ and } k_2 > 0, k_j < M_j, k_2 \geq w_2 \quad \forall j \in C \setminus \{1, 2\} \\ \mu_i k_i p_{id} & \text{if } s^* = (k_1, \dots, k_i - 1, \dots) \text{ and } k_i > 0, k_i \leq w_i \quad \forall i \in C \setminus \{1\} \\ \mu_i w_i p_{id} & \text{if } s^* = (k_1, \dots, k_i - 1, \dots) \text{ and } k_i > 0, k_i \geq w_i \quad \forall i \in C \setminus \{1\} \end{cases}$$

where all other transition rates, q_{ss*}, are 0.

All patients arrive at the first node of the network, and therefore only k_1 is subject to increase by a rate of $\lambda(\xi)$. Consider a case where $M_1 = 10$. Then the transition $s = (k_1, k_2, k_3, k_4, k_5) = (5, 10, 2, 3, 2) \rightarrow s^* = (\mathbf{6}, 10, 2, 3, 2)$ occurs with a rate of $\lambda(\xi)$. Internal flows of the network occurs from either node 1 or 2, and all discharges from either node 2, 3, 4 or 5. These are all dependent on both service rates, assigned staff and the routing probabilities of the node where the patient has just completed service. Thus if $M_3 = 5$ and $w_2 = 2$, $s = (6, 10, 2, 3, 2) \rightarrow s^* = (6, \mathbf{9}, \mathbf{3}, 3, 2)$ occurs with a rate of $\mu_2 w_2 p_{23}$, and if $w_3 = 4$, $s = (6, 9, 3, 3, 2) \rightarrow s^* = (6, 9, \mathbf{2}, 3, 2)$ occurs with a rate of $\mu_3 k_3 p_{3d}$.

Time-Dependent Behavior

To derive the waiting times from the queueing network, we do not only have to take the assigned staff into account, but also the effect of the time-varying arrival rate, as was defined in Section 5.2. The approach we follow could be classified as a piecewise transient model according to Schwarz et al., 2016 [113] and the solution method we apply is uniformization [123], also denoted randomization.

Notice that, as the arrival rate is weekly cyclical and *if* the working-patterns are weekly cyclical as well, then the process eventually stabilizes with the distribution given as a weekly-periodic vector function, $f(\xi)$. We make a numerical approximation to this distribution by first assuming that the change in arrival rate is negligible within some limited time interval, for instance, one hour. We denote the length of these time intervals by δ such that the length of the period, of one week, τ^{week} is an integer multiple of δ. Now, let $\lambda(\xi)$ define the arrival rate of patients at time ξ, and assume there exists a negligible change $|\lambda(\xi) - \lambda(\xi + \delta)|$, so $\lambda(\xi)$ can be discretized into a vector $\boldsymbol{\lambda}$ of size $\tau^{week}/\delta \in \mathbb{N}^+$.

Let $\pi_i(t)$ define the i'th segment of the stabilized distribution a function of t with $\{t \in \mathbb{R} | 0 \leq t \leq \delta\}$. Furthermore, let $\Upsilon = \{\xi \in \mathbb{R} | 0 \leq \xi \leq \tau^{week}\}$ and $t = 0$ represent the beginning of a segment, i, on the time line of Υ, and $t > 0$ the duration of time spent in such a segment, so $\pi_i(t)$ can be determined on any $\xi \in \Upsilon$ using both i and t, as illustrated in Figure 5.4.

Then, (5.3) represents the time-dependent state distribution of the process for any point in time of the week, $\xi \in \Upsilon$, where all entries in the vector function $f(\xi)$ are piecewise constant.

$$f(\xi) = \pi_i(t), \quad i = \lceil \xi/\delta \rceil \wedge t = \xi - (i-1) \cdot \delta \tag{5.3}$$

where $\pi_{i-1}(\delta) = \pi_i(0)$.

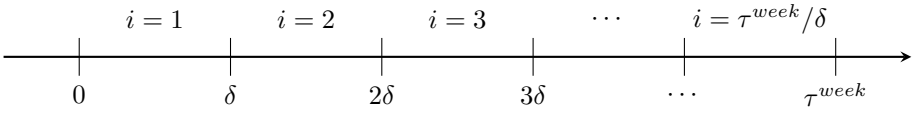


Figure 5.4: The segmented time-line of length τ^{weeks}. Applied in the modelling of the time-dependent state distribution, $f(\xi)$.

For each of the τ^{week}/δ time-intervals, illustrated in Figure 5.4, we require a method for deriving the time-inhomogeneous state distributions, $\pi_i(t)$, from which we derive the entire weekly behavior of the process.

From standard theory we have,

$$\pi_i(t) = \pi_i(0)e^{Q_i t} \tag{5.4}$$

where Q_i is the transition rate matrix containing the element λ_i from $\boldsymbol{\lambda}$ corresponding to the i'th segment. We use *uniformization* as presented below to calculate the matrix exponential.

Let γ_i be at least as large numerically as the largest diagonal element of Q_i. We then write,

$$P_i = Q_i/\gamma_i + I \tag{5.5}$$

where P_i is a transition probability matrix, with each element defining the probability of going from a state $s \in S$ to a new state $s^* \in S$, and I the identity matrix. Then, from an initial distribution $\pi_i(0)$, the distribution at time t, in segment i, $\pi_i(t)$, can be derived using (5.6) [123].

$$\pi_i(t) = \sum_{k=0}^{\infty} \pi_i(0)P^k \frac{(\gamma_i t)^k}{k!} e^{-\gamma_i t} \tag{5.6}$$

The transformation of Q_i into P_i, allows us to interpret our CTMC as an embedded Markov chain with a random number of transitions. The number of state changes in the embedded Markov chain, P_i, has probability $e^{-\gamma_i t}(\gamma_i t)^k/k!$ according to a Poisson distribution with parameter $\gamma_i t$, and thus depends on the time t. From $\pi_i(0)P^k$, the state distribution after exactly k changes is determined, so by using $e^{-\gamma_i t}(\gamma_i t)^k/k!$ we may weigh each of these distributions, according to time t and form the resulting state distribution $\pi_i(t)$. In implementing (5.6), we computationally use a recursive formulation to approach $\pi_i(t)$ until convergence.

Let K be the minimum number of terms in (5.6) required to attain an accuracy of ϵ, then the following statement has to be satisfied [123]:

$$\sigma_K = \sum_{k=0}^{K} \frac{(\gamma_i t)^k}{k!} \geq (1-\epsilon)e^{\gamma_i t} \tag{5.7}$$

From (5.7), a recursive formulation to determine K can be established – presented below:

1. Initialize setting $\zeta \leftarrow \sigma \leftarrow 1$ and $K \leftarrow 0$.
2. If $\sigma \geq (1-\epsilon)e^{\gamma_i t}$, then **stop**; otherwise continue.
3. Set $\zeta \leftarrow \zeta \frac{\gamma_i t}{K+1}$ and $\sigma \leftarrow \sigma + \zeta$.
4. Set $K \leftarrow K+1$ and go to **2**.

The stabilized distribution of the segment i, $\pi_i(t)$, can then be computed by using the recursion:

1. Initialize setting $y \leftarrow \pi \leftarrow \pi_i(0)$, and $k \leftarrow 0$.
2. If $k = K$, then **stop** as $\pi_i(t) \approx e^{-\gamma_i t}\pi$ with an accuracy of ϵ. Otherwise set $k \leftarrow k + 1$ and continue.
3. Set $y \leftarrow y\left(P_i \frac{\gamma_i t}{k}\right)$, $\pi = \pi + y$ and go to **2**.

This concludes the approach we use to derive the state distribution at time t in segment i, $\pi_i(t)$. Our implementation of the step function, $f(\xi)$, containing the state distribution at every point in time of the week, is derived recursively by using Algorithm 8, as is presented in the following Section 5.3.1.

Waiting Times

In the above, we have presented an approach to obtain the occupancy distribution for a duration of time t in a segment i, denoted $\pi_i(t)$, given the initial value $\pi_i(0)$. The entries of the transition rate matrix in each segment, Q_i, is determined by $\lambda(\xi)$, along with the allocation of staff to each queue of the network. Through (5.3), $\pi_i(t)$ is used to determine the state distribution for the entire week, $f(\xi)$. Though $f(\xi)$ specifies how many patients are expected to be present in each queue, the measure does not directly reflect the resulting waiting times.

Let W denote the waiting time at a queue with w servers, and let k define the number of patients present at the queue at the time of arrival. Then $W = 0$ if $k \leq w - 1$. For exponential service times and $k \geq w$ we have $W = \sum_{i=w}^{k} Z_i$, where Z_i are independent exponential random variables of rate w times the service rate of each server. Our aim is to derive the fraction of patients waiting below a specific target as function of time of the week, ξ. That is, $L(\xi) = Prob\{W(\xi) \leq \nu\}$, where ν is the target upper waiting time and $W(\xi)$ the time-dependent waiting time distribution. Letting $K(\xi)$ define the random number of patients present at the queue at time ξ, we assume the time-inhomogeneous behavior within a segment is negligible so,

$$Prob\{W(\xi) \leq \nu\} = \sum_{k=0}^{M_c} Prob\{W(\xi) \leq \nu | K(\xi) = k\} \cdot Prob\{K(\xi) = k\} \quad (5.8)$$

and therefore, by letting $f_{ci}(\xi) = \sum_{j \in J} Prob\{s = (\dots, k_c = i, \dots)\}$ define the marginal time-dependent state distribution obtained in Section 5.3.1, where $J = S \setminus \{s = (\dots, k_c \neq i, \dots)\}$ — that is, the probability that queue $c \in C$ is occupied by i patients, we get

$$Prob\{W_c(\xi) \leq \nu_c\} = \sum_{i=0}^{w_c - 1} f_{ci}(\xi) + \sum_{k=w_c}^{M_c} f_{ck}(\xi) \cdot Prob\left(\sum_{i=1}^{k} z_i \leq \nu_c\right) \quad \forall c \in C \quad (5.9)$$

from [78]. The first term of (5.9) accounts for the probability that there is no waiting time on arrival – namely when at least one of the servers is free. The second term contains the probability that the queue is occupied by w_c or more patients, and the probability that the sum of service times for these patients is equal to or less than the queue dependent target ν_c. Furthermore, as

$$Prob\left(\sum_{i=1}^{k} z_i \leq \nu_c\right) = \int_0^{\mu_c w_c \nu_c} \frac{u^{k-1}}{(k-1)!} \cdot e^{-u} du = 1 - \sum_{j=0}^{k-1} \frac{(\mu_c w_c \nu_c)^j}{j!} \cdot e^{-\mu_c w_c \nu_c} \tag{5.10}$$

this allows us to write (5.9) on the form,

$$L_c(\xi) = \sum_{i=0}^{w_c-1} f_{ci}(\xi) + \sum_{k=w}^{M_c} f_{ck}(\xi) \cdot \left(1 - \sum_{j=0}^{k-1} \frac{(\mu_c w_c \nu_c)^j}{j!} \cdot e^{-\mu_c w_c \nu_c}\right) \quad \forall c \in C \tag{5.11}$$

where $L_c(\xi)$ is the fraction of patients waiting for staff type $c \in C$ below the target ν_c at time ξ. Let t define an hour of the week in the set $T = \{1, 2, \ldots, 168\}$, then for the remaining of this study, we refer to (5.11) as the function $L_{tc}(\mathbf{Z})$, presented in the master problem (5.2a)-(5.2d). The time targets for staff type $c \in C$, ν_c, are presented in Appendix B.1, Table B.3. Finally, we apply (5.11) based on the time-dependent distribution, $f(\xi)$, using Algorithm 8.

Algorithm 8 Algorithm for evaluating the system over a full week.

$\pi_0 \leftarrow (1, 0, 0, \ldots, 0)^T$
$L_0 \leftarrow WAITINGTIME(\pi_0)$
while $d > tol$ **do** ▷ Run until tolerance is satisfied
  $i \leftarrow 1$
  while $i < 169$ **do**
    $\pi_i \leftarrow UNIFORMIZE(\pi_{i-1})$ ▷ Uniformize at the end of the i'th hour
    $L_i \leftarrow WAITINGTIME(\pi_i)$ ▷ Evaluate waiting times in network using (5.11)
    $i \leftarrow i + 1$
  end while
  $d \leftarrow RELATIVETOL(L_{168}, L_0)$ ▷ $d = max_{c \in C}(L_{168,c} - L_{0,c}))/L_{0,c}$
  $\pi_0 \leftarrow \pi_{168}$
  $L_0 \leftarrow L_{168}$
end while
return L

Notice, as we are only concerned with the instance where $t = 0$, we suppress the dependency on t, and let $\pi_i(0) = \pi_i$. Further, for convenience in Algorithm 8 we let L_i define a vector of the elements $L_{tc}(\mathbf{Z})$ for all $c \in C$ with time index t corresponding to the i'th segment.

We initialize the algorithm by an empty system, setting $\pi_0 \leftarrow (1, 0, 0, \ldots, 0)^T$. We then discretize to form $\tau^{week}/\delta = 168$ time-intervals – one for each hour of the week. Next, we evaluate the system in each time interval by uniformizing the process at the end of the hour, using the preceding hour as input. After all hours have been evaluated, the maximum relative difference in waiting time from the beginning, L_0, to the end of the week, L_{168}, is used as stopping criteria. Notice, for a slightly faster algorithm, the evaluation of L_i for the remaining segments of the week can be postponed until $d \leq tol$.

5.3.2 Optimization Heuristic

In Section 5.3.1 we have presented how to model ED patient flow using a continuous-time Markov chain (CTMC) and derive the time-dependent behavior of the system by recursively uniformize the model until convergence. This approach yields a complex non-linear relation between assigning staff and the resulting patient waiting time.

Now, consider again the master problem (5.2a)-(5.2d). Let b_{ct} define a lower bound on staff of type $c \in C$ in time period $t \in T$ which is required to respect (5.2b) and (5.2c). Then (5.2a)-(5.2d) may be re-written in the form of an Integer Linear Programming (ILP) problem presented in (5.12a)-(5.12c) – which we can solve in reasonable time by applying a standard commercial solver software.

$$\text{min.} \quad \sum_{c \in C} \sum_{j \in J} x_{cj} \tag{5.12a}$$

$$\text{s.t.}$$

$$\sum_{j \in J} a_{cjt} x_{cj} \geq b_{ct} \qquad \forall t \in T,\, c \in C \tag{5.12b}$$

$$x_{cj} \in \mathbb{N}_0 \qquad \forall j \in J,\, c \in C \tag{5.12c}$$

Still, we are faced with the problem of deriving b_{ct} in (5.12b), in order to respect the master problem. Sinreich et al. 2012, [119] presented two recursive heuristic algorithms, combining both simulation and mixed integer programming. Their simulation model, which is developed by Sinreich & Marmor, 2005 [118], accounts for many different patient pathways upon which their heuristics have been developed. Their overall approach is to minimize the length of stay for patients by recursively identifying and removing bottlenecks in the system. In our study, the representation of the ED is more simple, as we consider only a single class of patients and queues consisting only of one staff type. Our matheuristic approach reflects this representation by recursively assigning staff to the queues of the network constrained by a fixed set of working-patterns and the waiting time distributions evaluated by (5.11). We elaborate more on this matheuristic in the following section.

Recursive Bound Adaptation

In this section we present a matheuristic search procedure, where working-patterns and constraints on waiting time are incorporated in a recursive man-

ner until a solution is derived. The heuristic has two stages: First a solution is constructed by recursively solving (5.12a)-(5.12c) and evaluating the resulting solution through the CTMC. The progressing of this first step constructs a feasible solution to x^*_{cj} in a greedy manner. Next, in the second stage, the meta-heuristic approach known as *Tabu Search* (TS) is used to search for an improved solution by further minimizing $\sum_{c\in C}\sum_{j\in J} x_{cj}$. We refer to this optimization strategy as Recursive Bound Adaptation (RBA).

The first stage consists of two parts:

1. **Optimization**. Let b^k_{ct} be a lower bound on required staff of type $c \in C$ in time period $t \in T$ for iteration k. Initializing with $b^0_{ct} = \beta_{ct}$, solve the ILP problem (5.12a)-(5.12c).

2. **Evaluation**. Starting from the solution, x^*_{cj}, derive the resulting allocation of staff $c \in C$ in time period $t \in T$, through $z_{ct} = \sum_{j\in J} a_{cjt}x^*_{cj}$. Then, evaluate the waiting times $L_{ct}(\mathbf{Z}) \quad \forall t \in T, c \in C$ by using the CTMC. Let $U_c \subseteq T$ be the set of time periods for staff type $c \in C$ for which, (5.2b), the waiting time constraint is violated. That is, $L_{ct}(\mathbf{Z}) < \tau$. Then, if $U_c \neq \emptyset$, make the adjustment: $b^{k+1}_{ct} = 1 + \sum_{j\in J} a_{cjt}x^*_{cj} \quad \forall t \in U_c, c \in C$, and $b^{k+1}_{ct} = b^k_{ct} \quad \forall t \in T \setminus U_c, c \in C$. Then, go to step 1 to generate a new allocation of staff, using b^{k+1}_{ct} as the new lower bound for (5.12b). Otherwise, if $U_c = \emptyset$, **stop**.

This recursive procedure ensures to not only derive a feasible solution to the master problem, but additionally as $b^0_{ct} = \beta_{ct}$, and b^k_{ct} is subsequently increased by 1, only in the time periods where $L_{ct}(\mathbf{Z}) < \tau$, ensures that x^*_{cj} is derived based on a tight lower bound. This solution should, therefore, serve as a promising input for the second stage. Here, we use a classic TS heuristic structure consisting of a neighborhood adjacent to the current solution, $N(x_{cj})$, the admissible subset of the neighborhood, $\tilde{N}(x_{cj})$, as well as a "tabu list" L of length $|L| = l$.

Furthermore, we consider two variations of the neighborhood definition. In the first, a probabilistic set of pattern-staff pairs is chosen from the total set $J \times C$. Here, a fraction, p_f, of the pairs are already used in the solution x_{cj}. For each of the chosen pairs, a random number $r \in \{-1, 1\}$ is generated, so that the neighborhood to be tested is $x_{cj} + r$. In the remaining of this paper, we refer to this definition as *add-remove*.

In the second variation, a probabilistic set of pattern-staff pairs are chosen from the set $J \times C$ again. However, all pairs must be used in the solution x_{cj}. Then, instead of adding additional staff to the solution, we consider that staff can be moved to another pattern that may, or may not, be used by x_{cj} already. Thus, a *move* is defined by the change $x_{cj} - 1$ followed by $x_{ci} + 1$, where $j \in Z_c$ is the set of patterns that *is* used by staff type $c \in C$, and $i \neq j \in J_c$, where J_c is the set of all patterns that *can* be used by staff type $c \in C$. To make sure that $\sum_{c\in C}\sum_{j\in J} x_{cj}$ is minimized, *moves* are a fraction of size p_f of all

elements in the neighborhood, where the rest are pure removals, as in the first neighborhood definition. Thus, we refer to this definition as *move-remove*.

Lastly, in order to evaluate the elements of the neighborhood, let $y \in \mathbb{R}_+$ be a "large" number which defines the penalty of violating (5.2b), so that the total penalty a solution generates is $\sum_{c\in C}\sum_{t\in U_c} y(\tau - L_{ct}(\mathbf{Z}))$. The function from which we evaluate each solution in the neighborhood is then $\sum_{j\in J}\sum_{c\in C} x_{cj} + \sum_{c\in C}\sum_{t\in U_c} y(\tau - L_{ct}(\mathbf{Z}))$.

Our TS heuristic is presented in Appendix B.2. The overall structure of the RBA heuristic is presented in Algorithm 9.

Algorithm 9 The overall structure of the Recursive Bound Adaptation heuristic.

$b_{ct} \leftarrow \beta_{ct}$ ▷ Initialize
$x_{cj} \leftarrow SOLVE(b_{ct})$
$U_c \leftarrow EVALUATE(x_{cj})$
while $U_c \neq \emptyset \quad \forall c \in C$ **do** ▷ Adjust bound b_{ct} until x_{cj} is feasible cf. (5.2b)
    $b_{ct} \leftarrow 1 + \sum_{j\in J} a_{cjt} x_{cj} \quad \forall t \in U_c, c \in C$
    $x_{cj} \leftarrow SOLVE(b_{ct})$
    $U_c \leftarrow EVALUATE(x_{cj})$
end while
$x^*_{cj} \leftarrow x_{cj}$
while $elapsedtime < maxtime$ **do** ▷ Attempt to improve the solution by using tabu search
    $x^*_{cj} \leftarrow TABUSEARCH(x^*_{cj})$
end while
return x^*_{cj}

5.4 Results

In this section, we test and apply the continuous-time Markov Chain (CTMC), as well as the Recursive Bound Adaptation (RBA) matheuristic presented in Section 5.3. Firstly, we derive the truncation of the CTMC and evaluate the model by comparing to a simulation of the associated system. This is presented in Section 5.4.1. In Section 5.4.2 we demonstrate the RBA matheuristic by firstly tuning the parameter setting, and subsequently conducting optimizations experiments for a number of different input datasets. We then evaluate our approach by comparing to a simulation, taking all patient classes into account.

5.4.1 Evaluation of the CTMC Model

Recall from Section 5.3.1 that our modeling approach assumes a finite upper bound, $M_i \; \forall i \in C$, limiting the number of patients that can be contained in the system at each queue in the network. To decide on a setting of these $|C|$

M_1	M_2	M_3	M_4	M_5	Tolerance	Runtime (s)
27	62	22	10	4	$5 \cdot 10^{-2}$	1254.7
63	104	35	14	6	$1 \cdot 10^{-2}$	32716.8
63	104	35	15	6	$5 \cdot 10^{-3}$	34992.8

Table 5.1: Results from adjusting the limit $M_i \; \forall i \in C$. Shows the resulting parameter setting, probability tolerance used in each test, and the runtime associated evaluating the system.

parameters, we conduct a number of tests, where we gradually increase each parameter in sequence until the maximum probability of attaining the bound respects a predetermined tolerance. In increasing M_i, we choose a sequence going downstream the network, such that the parameter for all potential upstream queues are determined. Furthermore, we realize that as the marginal state probabilities depend on the load of the system, so does the appropriate setting of M_i. For this reason, we conduct our tests using the arrival rate $\boldsymbol{\lambda}$ and the lower bound β_{tc}, cf. the optimization problem (5.2a)-(5.2d), yielding the largest load that will ever be encountered by the system. For the remaining of this study, we choose β_{tc} such that $\beta_{tc} = \left\lceil \frac{\lambda_t p_{ic}}{\mu_c(1-p_{cc})} \right\rceil \; \forall t \in T, c \in C$, where p_{ic} defines the probability of a patient going to queue c from the predecessor i. Notice that this definition ensures the minimum number of servers that prevents an over-utilized system for each segment of the time-line separately.

We conduct our tests using three different tolerance levels. The resulting setting of each $M_i \; \forall i \in C$ and the runtimes associated with evaluating the system, is presented in Table 5.1. Here we notice that both the required state space, as well as the associated runtime, increase excessively, despite the fairly limited size of the state space. This would indicate that the system can become computationally intractable if the arrival rate increases, or a small tolerance is required to attain sufficient accuracy.

Now, in our subsequent experiments we demonstrate that using the setting $M_1 = 27$, $M_2 = 62$, $M_3 = 22$, $M_4 = 10$ and $M_5 = 4$ is adequate. We conduct these experiments by comparing the marginal state distributions, and waiting times as they were defined in (5.11) to a discrete-event simulation of the CTMC behavior. We conduct these experiments using two different staff profiles for which we fix the number of servers over the entire week in each queue. Furthermore, we assess the model sensitivity to the assumption that service times are exponentially distributed by comparing to simulations where service times follow a log-normal distribution.

Our simulation model was implemented using the modeling language Matlab, and all experiments were conducted for a simulation time of 416 weeks (8 years) including 8 weeks of burn-in. An overview of all simulation experiments are presented in Table 5.2, showing the service time distribution and the staff profile used in each run.

	Service Time		Servers				
#	Distribution	Standard Dev.	w_1	w_2	w_3	w_4	w_5
1	Exponential	$\sigma_c = 1/\mu_c$	1	3	3	2	1
2	Log-normal	$\sigma_c = 1/\mu_c$	1	3	3	2	1
3	Log-normal	$\sigma_c = 2/\mu_c$	1	3	3	2	1
4	Exponential	$\sigma_c = 1/\mu_c$	2	4	3	2	2
5	Log-normal	$\sigma_c = 1/\mu_c$	2	4	3	2	2
6	Log-normal	$\sigma_c = 2/\mu_c$	2	4	3	2	2

Table 5.2: Overview of the simulation experiments used to assess the CTMC model adequacy. Shows the service time distribution and the number of servers used in each run.

The results were evaluated by graphically comparing the two measures. We assessed the marginal state distributions on the expected state, according to $f_{ci}(\xi)$, by sampling the system state at the beginning of each hour in the simulation period. Further, the waiting time service level was evaluated by sampling waiting times on arrival to the respective queues, and then deriving the fraction corresponding to $L_c(\xi)$ from the resulting distributions. Examples of the experiments from Table 5.2 are presented in Figure 5.5, showing the waiting time service level in experiment 1, 2 and 3 for queue 1 (triage) and queue 5 (orthopedic surgeons), respectively.

For the experiments where simulation is compared directly to the CTMC (1 and 4, cf. Table 5.2), we find that the difference in expected state, as well as waiting time service level, is fairly negligible in all cases. Moreover, when we adjust the service time distribution to log-normal, the change is only distinct in the cases where standard deviation is twice the expected service time. Furthermore, Figure 5.5 demonstrates that the system is dependent on the service time distribution, but that the sensitivity depends greatly on the queue in focus.

5.4.2 Evaluation of the RBA Heuristic

In order to demonstrate our heuristic search procedure we have defined a range of datasets consisting of the interaction between the three arrival patterns, $\Lambda = \{\lambda(\xi) \cdot 0.9^2, \lambda(\xi) \cdot 0.9, \lambda(\xi)\}$ (cf. Section 5.2.1), along with three waiting time service levels, τ, $\mathcal{T} = \{0.7, 0.8, 0.9\}$, cf. (5.2b). Together, these make up nine different datasets, presented in Table 5.3.

To determine the appropriate parameter setting for our subsequent optimization experiments, we apply the RBA heuristic to the following datasets: `Low70`, `Medium80`, `High90` which essentially represent three different levels of load to the system.

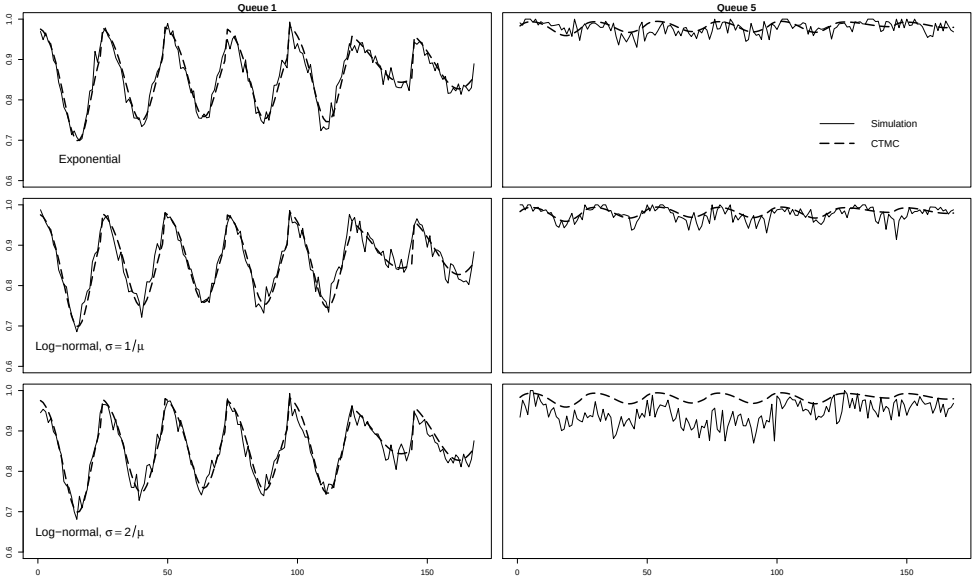


Figure 5.5: The waiting time service level as function of week-hour. Compares the CTMC model and simulation, on experiment 1, 2 and 3 (cf. Table 5.2), and queue 1 and 5, respectively.

Reference	Low70	Medium70	High70	Low80	Medium80	High80	Low90	Medium90	High90
Λ	$\lambda(\xi) \cdot 0.9^2$	$\lambda(\xi) \cdot 0.9$	$\lambda(\xi)$	$\lambda(\xi) \cdot 0.9^2$	$\lambda(\xi) \cdot 0.9$	$\lambda(\xi)$	$\lambda(\xi) \cdot 0.9^2$	$\lambda(\xi) \cdot 0.9$	$\lambda(\xi)$
$\mathcal{T}$	0.7	0.7	0.7	0.8	0.8	0.8	0.9	0.9	0.9
Used in	Tuning	Testing	Testing	Testing	Tuning	Testing	Testing	Testing	Tuning

Table 5.3: Datasets used in parameter tuning and testing of our two heuristic approaches.

Let z_d^* define the best known solution for dataset $d \in D$, where $D = \{\texttt{Low70}, \texttt{Medium80}, \texttt{High80}\}$. Then, the performance of each specific parameter setting is evaluated for dataset d, by using the average percentage gap, E_d, and variance, σ_d^2, presented in (5.13a)-(5.13b),

$$E_d = \frac{1}{N} \sum_{i=1}^{N} \frac{z_i - z_d^*}{z_d^*} \cdot 100\% \qquad \sigma_d^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - E_d)^2 \qquad \text{(5.13a, 5.13b)}$$

where z_i and $x_i = (z_i - z_d^*)/z_d^*$ are the resulting fitness and percentage gap of replication $i \in \{1, 2, 3\}$, respectively. Further, to determine the overall performance of each of the tested parameter settings we let E_{tot} define the overall average percentage gap, and σ the pooled standard deviation, presented in (5.14a)-(5.14b).

$$E_{tot} = \frac{1}{n_D}\sum_{d=1}^{n_D} E_d \qquad \sigma = \sqrt{\frac{\sum_{d=1}^{n_D}(N-1)\sigma_d^2}{n_D(N-1)}} \qquad \text{(5.14a, 5.14b)}$$

We conducted a full interaction test adjusting the penalty y on the levels 40 and 10000, and the fraction p_f on the levels 0.25 and 0.75, respectively. The remaining parameters were fixed based on preliminary testing. The experiments were again conducted on the two variations of the heuristic. That is, the *add-remove* and *move-remove*. Each setting was replicated twice for each of the three datasets with a time-limit of 10 hours.

Running these experiments, none of the settings were able to improve the solution subsequent to the first recursive stage of the heuristic. For this reason, we have chosen an arbitrary parameter setting, presented in Appendix B.1, for our later optimization experiments.

Optimization Experiments

The optimization experiments were conducted on the remaining six datasets: `Medium70`, `High70`, `Low80`, `High80`, `Low90` and `Medium90`. Each run of the RBA heuristic was replicated three times using af fixed setting of **24 hours of runtime** in the second stage. The ILP problem in (5.12a)-(5.12c) was solved by using the IBM ILOG CPLEX Optimizer.

The results for each dataset and variation of the heuristic are presented in Table 5.4, showing the total amount of staff that is initially derived by the first recursive stage, and subsequently by the second TS stage. The latter is presented in three columns which contains the results obtained in each replication of the heuristic, whereas the first stage is presented in a single column due to its deterministic progression.

Now, our experiments show that the TS variations produce similar and quite consistent results. Regarding the difference between the different datasets, the amount of allocated staff is clearly sensitive to the arrival rate and the specified service level targets. The ILP problem in (5.12a)-(5.12c) was solved in less than 10 seconds for all cases, and with 2-6 iterations in the first stage. Moreover, the input for the second TS stage turns out to improve in only a single case, indicating that the first stage returns solutions that are close or exact optimums, *or* since the optimal solution is unknown it may also be the case that our second TS stage implementation is inefficient.

Solution Evaluation

Now recall from Section 5.2 that the ED is in fact subject to four different patient classes determining the order in which patients are prioritized. To assess whether the solutions derived in the preceding section has any implications for a system incorporating all four classes, we applied each solution to our discrete-event simulation model from Section 5.4.1 by distinguishing between

	First Stage			Second Stage					
				TS add-remove			TS move-remove		
Dataset	Allocated Staff	Iterations	Runtime (s)	1	2	3	1	2	3
Medium70	33	2	3017	33	33	33	33	33	33
High70	35	3	4496	35	35	35	35	35	35
Low80	33	2	2839	33	33	33	**32**	33	33
High80	39	3	4571	39	39	39	39	39	39
Low90	40	5	8079	40	40	40	40	40	40
Medium90	42	6	9826	42	42	42	42	42	42

Table 5.4: Results from testing the RBA heuristic on the remaining datasets. Shows both the solution that is derived in the first stage of the heuristic, and the subsequent (replicated) TS solution.

patient classes, as well as including the possibility of changing priority subsequent to the second queue.

Once again, our simulation experiments were conducted using a simulation time of 416 weeks including 8 weeks of burn-in. In order to depict the general implications for each respective patient class, we have derived the waiting time service level as an average, *weighted* according to the number of patients arriving at each queue over time.

The results are presented in Table 5.5 showing each patient class and dataset, respectively. As expected, the service level is increasing in accordance with both the priority of each class and the target of each dataset. Furthermore, we find that the service level is always attained above the target for patients of level 3 and 4, but slightly violated in a single case on level 2 and half of the cases on level 1. In this regard, note that for some ED cases is the service level dependent on the triage level, often yielding a less ambitious waiting time target for patients of lower priority.

Triage / Datasets	Level 1	Level 2	Level 3	Level 4
Medium70	0.801	0.839	0.907	0.970
High70	0.814	0.861	0.917	0.973
Low80	0.787	0.832	0.900	0.967
High80	0.880	0.916	0.968	0.986
Low90	0.876	0.906	0.984	0.994
Medium90	0.855	0.889	0.959	0.984

Table 5.5: The simulated waiting time service level for each patient class. Presented as an average, weighted according to the amount of patients arriving at each queue over time.

5.4.3 Discussion

Through Section 5.3 and 5.4, we have presented and tested an approach for modeling ED patient flow based on a CTMC, which accounts for the time-

dependency in the system resulting from a realistic time-varying arrival rate of patients, and presence of staff. Even though we capture many of the essential elements of an ED, is our model based on a few simplifications such as assuming that service times are exponentially distributed. We have further noted that our flow system does not incorporate the extensive structure as has been considered by related simulation studies [118, 82]. However, simulation experiments have indicated that our CTMC is robust to the service time distribution, and derives an accurate state distribution faster than the associated simulations. We have further found that our approach, considering only a single merged class of patients, is adequate in evaluating the performance of the associated multi-class system.

For the optimization of the system, we have been investigating a matheuristic approach on a number of different input datasets. The approach consists of firstly a recursive stage, where the lower bound on staff is greedily increased until the constraint on waiting time is respected. As allocating staff to one period affects the service level in all other time periods, optimality cannot be guaranteed by using this procedure. A TS heuristic is, therefore, added to search for any "excess" staff.

The advantage of this procedure is the greedy adaptation of a lower bound, initialized at its lowest possible level, and therefore inclined to produce a promising solution. On the other hand, the approach faces the problem of repeatedly evaluating both the CTMC and ILP problem, which can be computationally expensive for more complex cases. For our case the problem of assigning staff to a limited set of working-patterns can be solved in below 10 seconds, and for this reason the RBA heuristic is able to derive a feasible solution within a reasonable amount of time from the first stage of the heuristic alone.

Lastly, an important question remains as to how far the obtained solutions are from the true optimum. A time-dependent queueing network makes up a range of dependencies, such as the load-dependency between queues in the network, and the effect that one time period has on all other time periods due to the weekly cyclic behavior. There is to our knowledge no standard method of deriving an optimality gap in a system that comprises these relations that does not involve an exhaustive evaluation of all permutations for a fixed sum $\sum_{c \in C} \sum_{j \in J} x_{cj}$, which can be quite computationally expensive, as we have demonstrated earlier. However, recall that results from the second stage in Table 5.4 could indicate that our solutions are near-optimal, since there is only improvement in a single case.

5.5 Conclusion & Future Work

In this study, we have aimed at providing a continuous-time Markov chain (CTMC) approach for the modeling of time-varying behavior of patient waiting time, and the interaction of this approach with an Integer Linear Programming (ILP) model. We have tested a matheuristic approach to the problem of allo-

cating staff to an Emergency Department (ED) which we refer to as Recursive Bound Adaptation (RBA).

In the literature, we have found that a range of different methods is used in patient flow modeling, but only a few of these studies considered optimizing the system. Even fewer studies have explored modeling and optimizing the ED based on a time-dependent queueing network. In our study, we have modeled time-dependency by discretization of the patient arrival rate and defining a step function of consecutive uniformizations of the CTMC. By conducting numerous simulation experiments, we have found that this approach is adequate for modeling the system occupancy, as well as waiting time, and is fairly robust to adjustments in the service time distribution.

By applying the CTMC to our matheuristic approach, we provide solutions that satisfy targets on patient waiting time, when we reduce the system to that of only a single class of patients. Further simulation experiments have shown show that these solutions perform well in an associated multi-class system, with only slight violations for the least prioritized patients.

Our model approach has been based on the essential elements of acute patient flow, which might be insufficient for other hospital cases. However, with this study, we have provided a method that adequately evaluates patient waiting times, which do not rely on sampling, and is therefore suitable in the context of optimization. Moreover, we deem that the approach presented in this study, may serve as a basis for further exploration within the area of ED optimization. Finally, the reader should notice that our matheuristic is not only limited to the specific system nor ILP that has been tested in this study, but can be used for other similar cases.

5.5.1 Future Work

The approach that was presented in this study provides a range of different aspects to consider in future work. The recursive first stage of the RBA heuristic could be diversified by the use of a restricted candidate list, as in the well-known Greedy Randomized Adaptive Search Procedure (GRASP). Furthermore, over-allocation of staff may be avoided by adjusting time periods of the bound sequentially.

Our study did not include any lower bounds to the master problem presented in (5.2a)-(5.2d) which is otherwise necessary to conduct a proper assessment of the solutions obtained by our matheuristic. For this reason, we deem that further work into exact solutions of the relaxed master problem should be considered.

Lastly, the CTMC have provided an analytical approach for evaluating time-dependent patient waiting time in an ED. Further patient data should be obtained to evaluate this modeling approach — for instance on patient waiting time and the service time distributions of each staff type. Moreover, analysis into larger flow systems should be studied to approach ED cases of more complex structure.

Acknowledgments

This research was supported and funded by Region Sjælland. The managing organization of seven public hospitals. Particularly, we thank the department of Production, Research and Innovation (Produktion, Forskning og Innovation) for their support in providing data and insight into the operations of the Danish hospitals, and Associate Prof. Anders Stockmarr for statistical advice.

Chapter 6

Simulation-based Rolling Horizon Scheduling for Operating Theatres[1]

Anders Reenberg Andersen, Thomas Jacob Riis Stidsen
and Line Blander Reinhardt

Abstract Daily scheduling of surgical operations is a complicated and recurrent problem in the literature on health care optimization. In this study, we present an often overlooked approach to this problem that incorporates a rolling and overlapping planning horizon. The basis of our modeling approach is a Markov decision process, where patients are scheduled to a date and room on a daily basis. By assuming that both state and action space is only partially observable, we apply our model in an on-line scheme known as rollout, where actions are constructed using a heuristic search procedure.

Our objective in this study is to test the potential of using this modeling approach on the resulting hospital costs, and number of patients that are outsourced to avoid violating constraints on capacity.

Using data from a Danish hospital, we find a distinct improvement in performance when compared to a policy that resembles a manual planner. Further analysis shows that substantial improvements can be attained by employing other simple policies, and a myopic heuristic search procedure.

6.1 Introduction

The hospital operating theatres are among the key elements of running a hospital involving a range of different clinical specialization from organ to orthopedic surgery. In recent years, the use of resources in this part of the hospital has received a substantial amount of attention in the Danish health care sector. In March 2015, the National Audit Office of Denmark [96] published a report on the use of staff resources based on four departments in orthopedic surgery with the conclusion that staff working hours are not ensured to be fully utilized for a majority of cases. Other governmental reports suggest a lack in resource utilization for the operating theatres as well. In September 2015, the Danish Ministry of Health [97] published a report on the overall status of the public health care sector, showing that the waiting time for surgery has been

[1] Submitted to Annals of Operations Research

increasing for 25% of the selected departments in the period of 2011 to 2014.

On top of the above, running hospital operating theatres is a quite complicated task. To ensure compliance with targets on patient waiting time, along with efficient use of both staff and equipment resources, decisions on multiple organizational levels have to be considered [64, 87]. These range from long horizon planning problems, such as deciding on the overall required capacity, to day-to-day scheduling (and re-scheduling) of patients for operation. Among the important elements in the scheduling of operating theatres, is the definition of a medium-term Master Surgical Schedule (MSS) defining the time-windows and rooms allocated to each of the clinical specializations. In the day-to-day scheduling of procedures to a specific time and room, the hospital planners are constrained by these "windows" defined in the MSS. Furthermore, we performed interviews and found that planners have to consider equipment constraints, rosters for the surgeons, overtime-costs, targets on both waiting time and utilization of rooms, and so on. Hence, the problem of scheduling patients for operation yields a time-consuming and complicated task for the hospital planners to overcome.

Our objective in this study is to provide hospital planners with a decision tool capable of optimizing the scheduling of patients for operation, respecting the constraints that are relevant to the planners. In our case, we consider that patients are scheduled on a day-to-day basis and require that a rolling and overlapping planning horizon is taken into account. Thus, the decisions that are made on each day have to be anticipative.

Our methodological approach will be a mathematical model, where we minimize the hospital costs resulting from a sequence of decisions by employing a Markov Decision Process (MDP) approach. The MDP is applied in a simulation-based rollout framework resulting in a heuristic policy. Our aim is to assess this modeling approach in a setup where scheduling is conducted on a daily basis.

In Section 6.2 we present the specific problem of this study. Next, in Section 6.3 we present our modeling and solution approach, divided into two parts. First, the model structure of the MDP is presented, and subsequently how we have modeled costs and the arrival of patients to the hospital. In Section 6.4 we apply our approach to data from a Danish hospital-case and assess the MDP performance by comparing to other scheduling methods by employing simulation. Finally, we present our conclusion and suggestions for future work in Section 6.5.

6.1.1 Literature Review

On a more general level the problem of operating theatre (OT) planning is a recurrent topic that has been covered by a substantial amount of papers. There exists several surveys on the subject of which some of the recent have been conducted by Cardoen et al., 2010 [32], Guerriero & Guido, 2011 [64],

May et al., 2011 [87], and lately Samudra et al., 2016 [109] where 137 journal papers on the subject of OT planning were found in the period of 2004 to 2014.

With respect to the organizational decision levels, Guerriero & Guido, 2011 [64] find that the studies can be classified into three categories: Strategic (long-term decisions), tactical (medium-term decisions), and operational (short-term decisions). May et al., 2011 [87] add further three decision levels denoted: Very long-term, very short-term, and contemporaneous. The decisions relevant to the very long-term are related to the layout of physical resources [130], such as the construction of operating rooms. Long-term decisions are related to patient flow patterns and assigning overall capacity to surgical groups [25, 126]. Medium-term decisions involve defining the Master Surgical Schedule (MSS), where the clinical specializations are assigned to specific rooms and time-windows [129]. On the short-term, the patient procedures are assigned to a specific time and room on a day-to-day basis, and on the very short-term and contemporaneous level, last-minute scheduling and re-scheduling is conducted [52, 31, 49, 92, 79, 20, 51, 106, 50, 121].

By focusing on the short-term operational level of OT planning, we have found a range of different approaches and problem structures. Studies can mainly be categorized into considering completely deterministic "off-line" problems [31, 52, 128, 141], to incorporating uncertainty features such as random procedure time [79, 49, 20] and disruptions caused by emergency demand [79, 51]. Surprisingly, we only encountered a single study on the allocation of patients which accounted for the uncertainty in future elective arrivals [106]. In total, Samudra et al., 2016 [109] shows that incorporating stochasticity constitutes more than half of the papers on OT planning.

The specific modeling approaches of short-term OT planning range from mathematical programming and heuristics [31, 52, 49, 20, 51, 128, 141] to Discrete-event and Monte Carlo simulation [50, 121], and further to a mixture of these [79]. For the purely deterministic cases Xiang et al., 2015 [141] and Van Huele & Vanhoucke, 2014 [128] combines the surgical scheduling problem with a staff rostering problem. Xiang et al., 2015 [141] develops a modified Ant Colony Optimization algorithm and tests the model by using both data from the literature and real data from a Chinese hospital. Van Huele & Vanhoucke, 2014 [128] approach the problem by using Mixed Integer Linear Programming (MILP) based on the most frequent objectives and constraints from the literature. In Fei et al., 2010 [52] and Cardoen et al., 2009 [31] the focus is more on the scheduling and sequencing of the surgical procedures. Fei et al., 2010 [52] use an approach comprising two phases where patients are firstly assigned a date by using a column-generation-based heuristic, and subsequently sequenced by using a hybrid genetic algorithm. Cardoen et al., 2009 [31] focus on the sequencing of procedures and develop MILP models which lead to either exact or heuristic solutions.

For the studies that incorporate uncertainty, Batun et al., 2011 [20] and Lamiri et al., 2008 [79] use Stochastic Programming (SP) to minimize the total cost of scheduling patients over a planning horizon. Specifically, Batun et al.,

2011 [20] develops a two-stage stochastic MILP and investigates the impact of parallel surgery processing and pooling operating rooms. Related hereto, Lamiri et al., 2008 [79] develops an SP model, and moreover a method combining Monte Carlo simulation and a MIP model to schedule elective patients within a specific planning horizon, and emergency patients on the same day of arrival.

Methods based on MILP modeling can in some cases become too inefficient as found by Erdem et al., 2012 [51], where a MILP model and Genetic Algorithm (GA) are developed to reschedule elective patients upon the arrival of emergency patients. For the MILP model, Erdem et al., 2012 [51] finds that a commercial solver is sufficient for only a limited "light" case, and therefore develops a GA to find solutions close to optimality for the more complex cases. In addition, Denton et al., 2007 [49] focus on heuristic methods for deriving the sequencing of patients in operating rooms, and find that a simple sequencing rule can be used to optimize both waiting time and overtime-costs.

As the above has shown, there is generally a large emphasis on random duration of procedures and the impact from emergency arrivals, but limited focus on overlapping planning horizons from unknown future elective arrivals. Range et al., 2016 [106] schedule elective patients based on a MILP model and solve the problem by using a column generation approach. Future arrivals are taken into account by measuring the expected number of future patients who cannot be scheduled for surgery. Practically, this feature is applied by a piecewise linearized function in the MILP formulation. Additionally, they assume that future patients are scheduled evenly over the potential days for surgery.

In this study, we consider the problem of finding a cost-optimal allocation of current patients as a sequential decision problem with finite overlapping planning horizons for each future decision epoch, and with infinite decisions over the entire model. Instead of assuming how future patients are generally scheduled, we estimate the long-term expected costs as a function of the action taken in the current decision epoch, and a range of heuristic scheduling policies for future patients. More specifically, we base our model on a Markov decision process (MDP) approach that, as to our knowledge, has not been considered in other studies within OT planning. Even though, MDP standard theory provides a number of exact "off-line" methods [104], we consider a problem instance for which the state and action space is only partially observable. Therefore, we derive solutions in an "on-line" manner, using a simulation-based approach known as rollout [37, 38, 21].

6.2 Problem Description

We consider a hospital case for which a planner schedules surgical operations on a day-to-day basis. The hospital may treat both elective and emergency patients for a range of different clinical specializations, but operating rooms

have been reserved for both patient types and for each respective area of specialization in advance.

We assume that all patients have an upper limit on waiting time from the moment a surgical operation is requested. Thus, due to uncertainty in procedure duration and inter-arrival times, capacity may in some instances be insufficient so that patients have to be outsourced — either to an internal or external location. In other studies, outsourced emergency patients may cause an elective procedure to be canceled and therefore re-scheduled for operation on a new date [51]. In our case, we assume that emergency resources are rarely insufficient so that we may limit our scope to the scheduling of elective patients only. We further focus on a single clinical specialization, and assume that resources are negligibly shared with other specializations.

Requests for surgical operation occur with continuous random intervals, but we assume that the hospital planner is able to save all requests and not allocate these until the end of each day. Requests from elective patients only occur on regular workdays so the hospital planner has to make a decision five times a week of fixed interval.

In finding an optimal schedule there are multiple objectives to consider. Patient treatment is part of a system where both quantitative and qualitative factors play a major role in running a hospital. Long waiting times and overcrowding of wards cause a decrease in both subjective and objective care quality [90, 66]. In addition, insufficient operating room capacity leads to expensive overtime-costs that are further added to the setup costs of preparing for an operation. Outsourcing is one way to avoid these problems, but yields a logistical cost, and the patient may not be treated by the preferred resource.

Our objective in this study is to test a new approach for a *cost*-optimized scheduling of surgical operations to specific rooms and days. We limit our scope to the expenditures related to overtime-costs and setup of the operating rooms. Furthermore, we assume an upper limit on patient waiting time, but otherwise the resulting waiting time of a schedule is not accounted for. We further consider a number of constraints related to surgeon, room and equipment availability. Lastly, we assume that outsourcing is only allowed when aiming to respect the limit on waiting time, and all other constraints. These specifications will be elaborated in the following Section 6.2.1.

For the remaining of this paper we will refer to scheduled surgical operations as *procedures*. All recently occurred and unscheduled procedures are referred to as *requests*. Days for which the number of requests is positive, such that the hospital planner must decide on an allocation of these, will be referred to as *allocation epochs*, and are not to be confused with the *decision epochs* [104].

6.2.1 Constraints & Dynamics of the Problem

As mentioned above, our aim is to derive a cost-optimized schedule for all requests that have arrived during the day, repeating this process for all future days. When a decision is made, the hospital planner considers a discrete planning horizon of total length, $H \in \mathbb{N}$, such that from the end of the current allocation epoch, $t \in \mathbb{Z}$, all days that are considered in the scheduling problem are $t+1, t+2, \ldots, t+H-1, t+H$.

Let $X \in \mathbb{N}_0$ be a random variable defining the total amount of requests received on an arbitrary day. Then for all days where $X > 0$ a scheduling problem has to be solved with a planning horizon that has been "rolled" accordingly. Let $\delta \in \mathbb{Z}_{>t}$ define the subsequent allocation epoch to t. Further, let Ω_i, where $|\Omega_i| = H$, define the specific set of days contained in the planning horizon of an allocation epoch, i. Hence, if $\delta < t + H$, then $\Omega_t \cap \Omega_\delta \neq \emptyset$, as illustrated in Figure 6.1.

Let R define a finite set of operating rooms available to the hospital, then the planner has to make a decision involving *both* the finite and discrete planning horizon, and the operating room resources in R. The feasibility in scheduling a procedure for a specific room, $r \in R$, depends on a predefined surgical schedule as well as other constraints which are presented in the following Section 6.2.1. Furthermore, all allocations may induce a cost from setting up the room or when procedures stretch into overtime. Our assumptions related to these costs will be presented in Section 6.2.1.

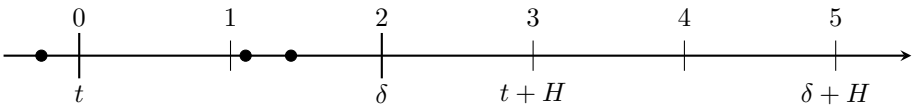


Figure 6.1: Example of a rolling and overlapping planning horizon of $H = 3$ days. Requests are illustrated by black dots along the time-line. As a result, planning has to be conducted at $t = 0$ and $\delta = 2$, leading to $\Omega_t = \{1, 2, 3\}$, $\Omega_\delta = \{3, 4, 5\}$ and $\Omega_t \cap \Omega_\delta = \{3\}$.

Constraints

Constraints relevant to the scheduling problem range from the availability of predefined capacity to less tangible factors such as preferences of the staff. In the below, we present each of these constraints separately.

1. **Number of rooms**. The hospital planner may decide to allocate requests to a number of rooms provided that an upper limit on open rooms is not violated. Let $y_{kl} \in \{0, 1\}$ be 1 if room $k \in R$ is being used on day $l \in T$, where $T = \{t+1, \cdots, t+H\}$ is the set of workdays within the current planning horizon; and otherwise 0. Further, let $c_l \in \mathbb{N}$ define the maximum number of rooms that is allowed to be opened on day $l \in T$. We assume that the structure of c_l is weekly cyclical such that $c_l = c_{l+5}$.

Then, an allocation to a room $k \in R$ on day $l \in T$ is only allowed if subsequently $\sum_{k \in R} y_{kl} \leq c_l$.

2. **Equipment**. Any procedure cannot be allocated to any room, even if c_l is not violated. To account for potential equipment requirements, as well as other preferences that may exist, each procedure type $i \in P$, where P is the set of all procedures that may occur, is constrained to a subset, U_i, of the available rooms, such that $U_i \subseteq R$.

3. **Physicians**. When a request is received by the hospital planner, a specific physician has already been assigned to conduct the procedure. We assume that physician-rosters are not flexible so requests can only be allocated to days for which the physicians are expected to be available at the hospital. Let J define the set of scenarios (or patterns) of days for which physicians will be available. As T is always a finite set, so is J. We assume that a request is randomly assigned to a specific pattern $j \in J$ with known probability.

4. **Opening hours**. Lastly, all operating rooms have a pre-specified time-interval for which they are expected to be open. Let $Y_i \in \mathbb{R}_{>0}$ be a random variable with known distribution that defines the duration of a procedure type, $i \in P$. Furthermore, let $r_{ikl} \in \mathbb{N}_0$ define the number of procedure $i \in P$ that are allocated to room $k \in R$ on day $l \in T$. Then, an allocation to a room $k \in R$ on day $l \in T$ is only allowed if there exists at least one sequence such that all procedures are expected to start within the opening hours. That is, $\sum_{i \in P \setminus \alpha}(r_{ikl} \cdot E[Y_i]) + (\sum_{i \in P}(r_{ikl}) - 1) \cdot m < w_k$, where $m \in \mathbb{R}_{>0}$ is a fixed buffer time, $w_k \in \mathbb{R}_{>0}$ is the time-capacity of room $k \in R$, and α is the allocated procedure with the longest expected duration for that room and day.

We assume that all allocations are *final* such that each respective procedure is locked in both room and date. Further, as neither of the above constraints are allowed to be violated, and that the occurrence of requests is independent of the current schedule, we allow for requests to be outsourced to yield a feasible solution with a maximum number of allocations. In this regard, we assume that allocating all current and future requests is always preferred over outsourcing any of them.

Costs

In combining a suitable schedule, the hospital planner has to consider that there might be a number of implications related to each respective solution. We found from interviews as well as from other studies [121] that some hospitals assess their performance on the utilization of time-capacity for each operating room. Such measure is convenient with respect to day-to-day monitoring and obtaining sufficient data, but does not provide an immediate relation between setting up new rooms and the risk of stretching procedures into overtime. For this reason, we evaluate the implications related to a specific

schedule on a sum of some different "penalties". We refer to these as *costs* as we mainly relate them to direct costs, such as overtime, cleaning, setting up equipment, and so forth. We have categorized these costs into two respective groups, presented below:

1. **Setup**. To account for the logistical costs related to equipment and staff preparation we assume that by opening a room the hospital receives a fixed *setup* cost. That is, the setup cost is induced only when the first procedures are allocated to the room, and does otherwise not depend on the utilization of time-capacity.

2. **Overtime**. As mentioned earlier, all procedures are subject to a random duration, and thus is in risk of stretching into overtime. If this is the case, we assume the hospital always pays a supplement to the staff independent of the type of procedure. In addition, some amount of discontent may arise among the staff leading to more errors and a decrease in the treatment quality. As a result we notice that the total penalty related to overtime could be a non-linear increasing function of the duration of overtime.

6.3 Modeling & Solution Approach

In this section, we present the approach we use to minimize the long-term expected costs of scheduling requests for operation. Our modeling approach is based on a Markov Decision Process (MDP) framework, for which, due to the problem size, we propose a simulation-based "on-line" solution method.

In Section 6.3.1 we present the specific structure of our modeling approach along with an exact solution method from standard theory. Next, in Section 6.3.2 we present our solution approach which is based on a simulation-based rollout method resulting in a heuristic policy.

6.3.1 A Markov Decision Process

Now, recall that we consider a finite set of procedures, $P = \{ProcedureA, ProcedureB, \cdots\}$. Any procedure, $i \in P$, are to be conducted within a fixed planning horizon, $H \in \mathbb{N}$, such that the set of future workdays in the planning horizon are in the set $T = \{t+1, t+2, \cdots, t+H\}$, where $t \in \mathbb{Z}$ is the day from which the planning horizon is observed. Further, let $R = \{Room\ A, Room\ B, \cdots\}$ define the total set of available operating rooms, and $r_{ikl} \in \mathbb{N}_0$ define the number of procedure $i \in P$ that have been scheduled on future day $l \in T$ in room $k \in R$.

In addition, we consider a finite set of all patterns for which physicians can be available within the planning horizon, $J = \{Pattern\ A, Pattern\ B, \cdots\}$. Together with P, these *availability-patterns* make up all of the attributes of any request that may occur. In other words, for any current day let $p_{ij} \in \mathbb{N}_0$ specify the number of requests of type $i \in P$ that are constrained by pattern $j \in J$.

Lastly, let $W = \{Monday, Tuesday, \cdots, Friday\}$ define the set of weekdays for which procedures can be allocated, and $d \in W$, then based on the above definitions, we introduce an MDP with state definition,

$$s = [p_{ij}, r_{ikl}, d] \tag{6.1}$$

divided into three parts: (1) The number and attributes of all current requests, p_{ij}, (2) the amount of each procedure scheduled to future day and room, r_{ikl}, and (3) the current weekday, d. Notice that d can be redundant depending on the structure of the problem from one case to another. If the constraints on room-capacity, c_l, and availability-patterns, J, can be generalized such that they are independent on the type of weekday in $l \in T$, then the state definition can be reduced to $s = [p_{ij}, r_{ikl}]$.

Furthermore, the reader should notice that the value of p_{ij} is generated by a purely stochastic process, whereas the transition into a state with any value of r_{ikl} will always be deterministic in terms of the decision by the planner. Now, let λ_i define the stationary occurrence rate of requests of type $i \in P$, and $X_{ij} \in \mathbb{N}_0$ be a random variable defining the occurrence of a request $i \in P$ constrained by pattern $j \in J$. Then the requests, X_{ij}, are generated according to a multivariate Poisson process with parameters $\lambda_{ij} = \lambda_i \xi_{ij} \quad \forall i, j \in P, J$. Here, $\xi_{ij} \in \mathbb{R}_{0<\xi_{ij}\leq 1}$ is the probability that a request of type $i \in P$ is constrained to pattern $j \in J$; hence $\sum_{j \in J} \xi_{ij} = 1 \quad \forall i \in P$.

In the following, we present how this modeling approach relates to the action space and transitions of the MDP.

Actions & Transitions

From one day to the next, the MDP changes from a current state $s \in S$ to a new state $s^* \in S$. This *transition* occurs consistently and with fixed time-interval. In addition, for each transition an action has to be chosen from the action space, A_s, available at each *decision epoch* — that is, at the end of every day, where the planner must decide on an allocation of the requests. Let π define a policy such that for any $s \in S$, $\pi(s) = \mathbf{a}$, where $\mathbf{a} \in A_s$. Thus for any arbitrary policy $\pi \in \Pi$, where Π is the set of all policies, the MDP will evolve as a Markov chain in discrete time.

Let $\mathbf{a}$ be a vector of the elements $a_{ijkl} \in \mathbb{N}_0$, defining the number of requests of type $i \in P$ constrained by pattern $j \in J$ that are allocated to room $k \in R$ on future day $l \in T$. To account for the outsourcing of requests we further extend $\mathbf{a}$ with the elements $q_{ij} \in \mathbb{N}_0$, defining the number of type $i \in P$ and pattern $j \in J$ that are outsourced. Thus, $\mathbf{a}$ has a total of $|P \times J \times R \times T| + |P \times J|$ elements. The size of A_s is, however, dependent on the values of r_{ikl} in the state s, which is limited by the constraints presented in Section 6.2.1. A_s contains any feasible value of $\mathbf{a}$; hence $1 \leq |A_s| \leq (|R \times T|)^{\sum_{i,j \in P,J} p_{ij}}$.

Notice that $\sum_{k,l \in R,T} a_{ijkl} + q_{ij} = p_{ij} \quad \forall i, j \in P, J$, and as the planning horizon is rolling $r^s_{ikl} + \sum_{j \in J} a_{ijkl} = r^{s^*}_{ik,l-1} \quad \forall i, k, l \in P, R, T \setminus \{t+1\}$, where r^s_{ikl} and $r^{s^*}_{ikl}$ are the schedules for the current state $s \in S$ and subsequent state $s^* \in S$, respectively. Moreover, notice that for $l = t + H$ all rooms are freed

such that $r_{ikl} = 0 \quad \forall i, k \in P, R$. However, as procedures are constrained to specific rooms, the only feasible solution may for some cases be to outsource all current requests. If for some decision epoch the number of requests $\sum_{i,j \in P,J} p_{ij} = 0$, then the only action is to let $\sum_{i,j,k,l \in P,J,R,T} a_{ijkl} + q_{ij} = 0$, in which case the MDP merely transitions into the next state resulting in $r^s_{ikl} = r^{s^*}_{ik,l-1} \quad \forall i, k, l \in P, R, T \setminus \{t+1\}$.

Lastly, the transition probability, $\mathbf{p}^{ss^*}_{\mathbf{a}}$, of changing from $s \in S$ to a subsequent $s^* \in S$ by choosing $\mathbf{a} \in A_s$, is merely $\mathbf{p}^{ss^*}_{\mathbf{a}} = Prob\{X_{11} = p_{11}, X_{12} = p_{12}, \cdots, X_{|P||J|} = p_{|P||J|}\}$ if $r^s_{ikl} + \sum_{j \in J} a_{ijkl} = r^{s^*}_{ik,l-1} \quad \forall i, k, l \in P, R, T \setminus \{t+1\}$; otherwise $\mathbf{p}^{ss^*}_{\mathbf{a}} = 0$.

Cost Function

In the previous section we introduced the policy $\pi \in \Pi$, where Π is the set of all possible policies for the MDP. Furthermore, recall that for any policy the MDP evolves as a discrete-time Markov chain. Let $V^{\pi}_{\infty}(s)$ define the expected long-term costs that are induced by this Markov chain, starting at state $s \in S$, under the policy $\pi \in \Pi$. That is,

$$V^{\pi}_{\infty}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t C(s_t, \pi_t(s_t)) \middle| s_0 = s\right] \tag{6.2}$$

where $C(s_t, \pi_t(s_t))$ is the cost induced from taking action $\pi(s_t)$ in state s_t at time t, $\gamma^t \in \mathbb{R}_{<1}$ a discount factor, and $t = 0$ is any arbitrary point in time. We define the optimal policy, π^*, as the policy which obtains $V^{\pi^*}_{\infty}(s) = \min_{\pi \in \Pi} V^{\pi}_{\infty}(s) \quad \forall s \in S$, and thus an essential element in minimizing the expected long-term costs is the definition of how each action is penalized through the cost function, $C(s_t, \pi_t(s_t)) = C(s, \mathbf{a})$. The reader should notice that the optimal myopic solution to the scheduling problem is included in the set Π, and thus we have that $V^{\pi^*}_{\infty}(s) \leq V^{\pi^{\eta}}_{\infty}(s)$, where π^{η} is the policy for which $\pi^{\eta}(s) = \arg\min_{\mathbf{a} \in A_s} E[\gamma^0 C(s, \mathbf{a})] \quad \forall s \in S$.

Now recall that we consider two different types of costs:

1. A fixed setup cost, $\kappa \in \mathbb{R}_{>0}$, is induced whenever a procedure is scheduled to a new room — that is, whenever $\sum_{i \in P} r_{ikl} = 0$ and $\sum_{i,j \in P,J} a_{ijkl} > 0$ for any $k \in R$ and $l \in T$ in the current state, s.

2. An overtime-cost that accounts for procedures stretching into overtime for any $k \in R$. Let the total capacity utilization of a room be defined by $\tau \in \mathbb{R}_0$, and let $f(\delta)$ define the overtime-cost for an overtime of size $\delta \in \mathbb{R}_0$, where δ is the amount of time that τ exceeds the capacity, w_k, for a room $k \in R$. Now, let $p_k(\tau)$ define the probability density function for a capacity utilization of amount τ in room $k \in R$. We then penalize an action according to the total *expected* amount of overtime, $\sum_{k \in R} o_k$, for the subsequent day, $l = t + 1$, where o_k is defined in (6.3). Notice that this formulation generalizes to any continuous distribution, $p_k(\tau)$, for which $\tau \geq 0$ and overtime-cost function, $f(\delta)$, for which $\delta \geq 0$.

$$o_k = \int_{w_k}^{\infty} p_k(x) f(x) \cdot dx \tag{6.3}$$

To ensure that actions are penalized for outsourcing requests, we further introduce a large penalty, $\phi \in \mathbb{R}_{>0}$ for every outsourced request. Finally, the resulting cost function is presented in (6.4), where y_{kl}^{s} and $y_{kl}^{s^*}$ is 1 if a room $k \in R$ is scheduled for use on day $l \in T$ in the current state $s \in S$ or subsequent state $s^* \in S$, respectively; and otherwise 0.

$$C(s, \mathbf{a}) = \sum_{k \in R} o_k + \sum_{k,l \in R, T \setminus \{t+H\}} (y_{kl}^{s^*} - y_{k,l+1}^{s}) \cdot \kappa + \sum_{i,j \in P,J} q_{ij} \cdot \phi \tag{6.4}$$

Exact Method

In Puterman, 2005 [104] a number of methods are presented for solving both finite and infinite horizon MDP problems. These include the algorithms *Value Iteration* and *Policy Iteration* that are used to approach an optimal policy. To our knowledge, one of the most widely used algorithms is Value Iteration for which a unichain and average reward approach is presented below:

Firstly, we let $\epsilon \in \mathbb{R}_{>0}$ define a tolerance such that π^* is the ϵ-optimal policy to the MDP. Secondly, let $\mathbf{v}^n$ define a vector of size $|S|$ with elements containing the *values* of each state $s \in S$ at iteration $n \in \mathbb{N}_0$. Further, let $sp(\mathbf{v}^n - \mathbf{v}^{n-1})$ define the "span" between two subsequent iterations, where $sp(\mathbf{x}) = \max_{i \in I} x_i - \min_{i \in I} x_i$ for any vector $\mathbf{x}$. An ϵ-optimal policy is then derived by using Algorithm 10, proposed by Puterman, 2005 [104].

Algorithm 10 The Value Iteration algorithm for an infinite horizon MDP.

Select v^0, ϵ ▷ Initialize
$n \leftarrow 0$
$span \leftarrow \infty$
while $span > \epsilon$ **do** ▷ Iterate until convergence
    $n \leftarrow n + 1$
    for all $s \in S$ **do**
        $v_s^n \leftarrow \min_{\mathbf{a} \in A_s} C(s, \mathbf{a}) + \sum_{s^* \in S} \mathbf{p}_{\mathbf{a}}^{ss^*} v_{s^*}^{n-1}$
    end for
    $span \leftarrow sp(\mathbf{v}^n - \mathbf{v}^{n-1})$
end while
for all $s \in S$ **do** ▷ Get the corresponding actions
    $\pi^*(s) \leftarrow \arg\min_{\mathbf{a} \in A_s} C(s, \mathbf{a}) + \sum_{s^* \in S} \mathbf{p}_{\mathbf{a}}^{ss^*} v_{s^*}^{n}$
end for
    return π^*

Notice that even if Algorithm 10 can be proven to converge in a finite number of iterations, the algorithm requires a full enumeration of all states in S as

well as all actions in each A_s. In our case the size of a single state and especially the state space, S, can be *very* large, even for small problem instances. Assuming a rather limited case where physicians are always available such that $|J| = 1$, procedures are constrained to only one room, and further that $c_l = c_{l+1} \quad \forall l \in T$, leading to $s = [p_i, r_{il}]$, there are a total of $|P| + |P \times T|$ elements in each state. That is, for a case with merely $|P| = 10$ different procedures, and a planning horizon of $|T| = 20$ days, a single state is comprised of 210 elements. Additionally, by assuming a maximum number, n, of requests per type, $i \in P$, and a capacity limit, m, of procedures per day, the state space would have a total size of $|S| = (n+1)^{|P|} \cdot \left(\frac{1}{|P|!} \prod_{i=1}^{|P|} (m+i) \right)^{|T|-1}$ states — for which a direct implementation of Algorithm 10 is quite a challenge, computationally. Furthermore, recall that in the worst case the action space attains a size of $|A_s| = (|R \times T|)^{\sum_{i,j \in P,J} p_{ij}}$. For these reasons we assume that the MDP considered in this study will only be partially observable.

In the following Section 6.3.2 we present a simulation-based approach with the aim of deriving a heuristic policy to the MDP.

6.3.2 A Heuristic Approach

The method presented in this section is based on a rolling-horizon approach. That is, instead of deriving an optimal action $\pi^*(s)$ for each of the states $s \in S$, we rely on deriving a *good* action heuristically in an "on-line" fashion. Our approach is based on a rollout algorithm proposed by Bertsekas & Castañon, 1999 [21], and later extended to parallel rollout by Chang et al., 2004 [37].

Consider some arbitrary allocation epoch, t, in which the requests p_{ij} are scheduled. These requests will be constrained by the occupation of the procedures that are already in the schedule, r_{ikl}, and for any policy induce the long-term cost $V_\infty^\pi(s)$. Now consider an optimal policy, $\pi \in \Pi$, that has been derived for a finite model-horizon H'. Then as $H' \to \infty$, the policy $\pi \to \pi^*$ for the infinite case. The cost of such a policy would then be,

$$V_{H'}^\pi(s) = E\left[\sum_{t=0}^{H'} \gamma^t C(s_t, \pi_t(s_t)) \middle| s_0 = s \right] \tag{6.5}$$

quite similar to (6.2). We assume for the remaining of this paper that $\gamma^t = 1$ for $t = 0, 1, \ldots, H'$. From the definition in (6.5), we note that the decision a hospital planner has to take from the current state s, should be derived from the sum of first the current *known* cost $C(s, \mathbf{a})$, and second an expected long-term cost from a sequence of future actions. Thus, we let a *rollout* policy, π', be defined as the result of a sequence of actions that has been derived under (6.6),

$$\pi'(s) \in \arg\min_{\mathbf{a} \in A_s} \{ C(s, \mathbf{a}) + E[\tilde{V}_{H'-1}^\pi(f(s, \mathbf{a}, \omega))] \} \tag{6.6}$$

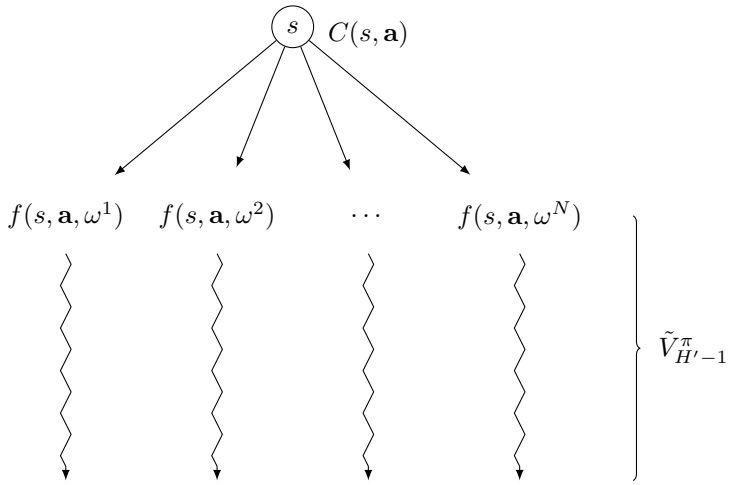


Figure 6.2: For the expression in (6.7), these are the subsequent paths and costs that are observed from the current state s.

where $E[\tilde{V}^{\pi}_{H'-1}(\cdot)]$ approximates (6.5), and $\tilde{V}^{\pi}_{H'-1}(\cdot)$ represents the total cost of a path of decisions over the horizon $t = 1$ to H'. Further, we let the subsequent state relative to s be defined as $s^* = f(s, \mathbf{a}, \omega)$. That is, the combined result of the current state, s, the action, $\mathbf{a}$, and a random disturbance of the system ω.

Similar to Bertsekas & Castañon, 1999 [21], we fix the disturbances, ω, to a finite set of values such that we limit our scope to a mere sample of the potential subsequent states. That is, we randomly sample N disturbances and then evaluate $\tilde{V}^{\pi}_{H'-1}(f(s, \mathbf{a}, w^j))$ for $j = 1, 2, \ldots, N$, yielding the N paths illustrated in Figure 6.2. Thus, for the decision of choosing an action in the rollout policy, π', (6.6) changes to (6.7).

$$\pi'(s) \in \arg\min_{\mathbf{a} \in A_s} \{C(s, \mathbf{a}) + \frac{1}{N} \sum_{j=1}^{N} \tilde{V}^{\pi}_{H'-1}(f(s, \mathbf{a}, \omega^j))\} \tag{6.7}$$

Notice that in practice, w^j is sampled by using pseudo-random numbers that are then converted into obtaining the requests, p_{ij}, at each subsequent state.

How to evaluate $\tilde{V}^{\pi}_{H'-1}(f(s, \mathbf{a}, \omega))$ will be presented in the following Section 6.3.2. Moreover, the expression in (6.7) requires a full enumeration of the state dependent action space A_s. As mentioned in Section 6.3.1, the size of A_s can be quite intractable, and therefore we require a robust search procedure to reduce the computational requirements. We present this procedure in Section 6.3.2.

Simulation-based Value Evaluation

Let Λ define a non-empty finite set of policies that all perform well for the hospital scheduling problem. By choosing the one policy that performs the best related to the current state, s, we allow for a rollout policy that continually adapts to the system. This is the basis of parallel rollout [37]. A related approach is to choose an action based on the current *weighted* average performance of the policies in Λ, which is the method we will employ in this study. We base our approach on a Simulated Annealing Multiplicative Weights (SAMW) algorithm proposed by Chang et al., 2007 [36]. Let $\phi(\pi)$ define the weighting of policy $\pi \in \Lambda$, such that $\sum_{\pi \in \Lambda} \phi(\pi) = 1$. The aim of the SAMW algorithm is then to concentrate the weighting on the *currently* (related to s) best policies in Λ.

Let $\phi^i(\pi)$ define the weight of policy π at iteration i. Then,

$$\phi^{i+1}(\pi) = \phi^i(\pi)\frac{\beta_i^{-\tilde{V}_i^\pi}}{Z^i} \tag{6.8}$$

where $\tilde{V}_i^\pi$ corresponds to $\tilde{V}^\pi_{H'-1}(f(s, \mathbf{a}, \omega^j))$ at iteration i for any of the disturbances ω^j. In addition, we have that $\pi \in \Lambda$ and $\beta_i \in \mathbb{R}_{>1}$ is a "cooling" parameter that decreases as function of iteration i. Furthermore, Z^i is a normalization parameter,

$$Z^i = \sum_{\pi \in \Lambda} \phi^i(\pi)\beta_i^{-\tilde{V}_i^\pi} \tag{6.9}$$

Now, we let $\omega_1^j, \omega_2^j, \ldots, \omega_{H'}^j$, where $\omega^j = \omega_1^j$, define a *path* of random disturbances such that we get $\tilde{V}_i^\pi = \sum_{t=1}^{H'} C(s_t, \pi(s_t))$, where $s_t = f(s_{t-1}, \pi(s_{t-1}), \omega_t^j)$, s_0 is the current state s, and $\pi(s_0)$ is the current action $\mathbf{a}$. In each iteration we generate a new range of disturbances (except for ω_1^j) and calculate $\tilde{V}_i^\pi \quad \forall \pi \in \Lambda$.

Letting $\mathcal{T}$ define a fixed number of iterations, we get the sample mean estimate $\psi(\pi) = \frac{1}{\mathcal{T}} \sum_{i=1}^{\mathcal{T}} \tilde{V}_i^\pi$ for each policy $\pi \in \Lambda$, which finally yields the approximation,

$$\tilde{V}^{\pi^*}_{H'-1}(f(s, \mathbf{a}, \omega^j)) = \sum_{\pi \in \Lambda} \psi(\pi)\phi^{\mathcal{T}}(\pi) \tag{6.10}$$

We use (6.10) to derive the last term of our rollout expression in (6.7). That is, (6.10) is used for each of the subsequent states that are illustrated in Figure 6.2. The overall structure of the SAMW algorithm is presented in Algorithm 11, where we predefine $\mathcal{T}$ experimentally to ensure a limited runtime of the algorithm. Moreover, notice that all disturbances, ω_t^j, can be generated prior to the running of Algorithm 11 as will be elaborated in Section 6.3.2.

The Search Procedure

Our approach for deriving an action, $\mathbf{a}$, from the current action space, A_s, is based on a Greedy Randomized Adaptive Search Procedure (GRASP). That

Algorithm 11 The Simulated Annealing Multiplicative Weights algorithm.

$\phi(\pi) \leftarrow 1/|\Lambda| \quad \forall \pi \in \Lambda$ ▷ Initialize the distribution
for $i = 1$ to $\mathcal{T}$ **do**
  $disturbances \leftarrow getNewDisturbances()$ ▷ New disturbances: $\omega_2^j, \omega_3^j, \ldots, \omega_{H'}^j$
  $\tilde{V}_i^\pi \leftarrow evaluate(disturbances, \pi) \quad \forall \pi \in \Lambda$
  $\phi(\pi) \leftarrow update(\phi(\pi), \tilde{V}_i^\pi) \quad \forall \pi \in \Lambda$ ▷ Update the distribution using (6.8)
end for
$\psi(\pi) \leftarrow average(\tilde{V}_i^\pi \quad \forall i) \quad \forall \pi \in \Lambda$
$\tilde{V}^{\pi^*} \leftarrow weightedAverage(\psi(\pi) \quad \forall \pi \in \Lambda, \phi(\pi) \quad \forall \pi \in \Lambda)$ ▷ Derive approximation using (6.10)
  return $\tilde{V}^{\pi^*}$

is, we conduct an iterative search consisting of two phases: (1) A greedy randomized solution, followed by (2) a local search procedure. We use this approach due to the combinatorial and greedy cost structure of the problem, ensuring that any immediate greedy allocation of requests will result in a reasonably low cost. The overall structure of the GRASP is presented in Algorithm 12.

Algorithm 12 General structure of the GRASP heuristic.

while $elapsedTime < maximumTime$ **do**
  $\mathbf{a} \leftarrow buildGreedyRandom(s)$ ▷ Construct greedy randomized solution from current state s
  $stop \leftarrow false$
  while $stop = false$ **do**
    $\mathbf{a} \leftarrow localSearch(\mathbf{a}, s)$ ▷ Try to improve the solution by local search
    $stop \leftarrow checkStoppingCriteria()$
  end while
end while
  return $bestFound(\mathbf{a})$ ▷ Return best solution from the entire search

For the greedy randomized solution, we generate a candidate list by enumerating all feasible allocations for each of the current requests, p_{ij}. Next, each of these allocations are ranked according to their apparent lowest cost increase. We then restrict this list to the $\alpha \in \mathbb{N}$ allocations with highest rank, and finally pick an allocation by random for insertion in the schedule, r_{ikl}. This process is conducted recursively until all requests have been allocated to the schedule.

For the ranking of each candidate allocation we conserve runtime for the later local search procedure, by only considering the current cost function, $C(s, \mathbf{a})$. Recall from (6.4) that the cost induced at every state is comprised of firstly a fixed setup cost, secondly an overtime-cost, and lastly a penalty for outsourcing requests. Thus, for an allocation to a room $k \in R$ on a day $l \in T$

we evaluate a candidate on the difference,

$$\Delta^i = o^i_{kl} - o^{i-1}_{kl} + (y^i_{kl} - y^{i-1}_{kl}) \cdot \kappa + q \cdot \phi \tag{6.11}$$

where Δ^i is the increase in cost *if* the i'th allocation is conducted for $i = 1, \ldots, \sum_{i,j \in P,J} p_{ij}$. Further, $o^i_{kl} \in \mathbb{R}_0$ and $y^i_{kl} \in \{0, 1\}$ is the overtime-cost and open-room indicator for the room $k \in R$ and day $l \in T$ for which the request is allocated, similar to (6.4). Notice that we can consider the cost *on* allocation, so $o^i_{kl} \geq o^{i-1}_{kl}$ and $y^i_{kl} \geq y^{i-1}_{kl}$. In addition, $q \in \{0, 1\}$ indicates if the request is outsourced; and κ and ϕ is the fixed setup cost and outsource penalty, respectively. Lastly, o^0_{kl} and y^0_{kl} are inherited directly from the current state, s.

Afterwards, the local search procedure intensifies the solution that has been created in the greedy randomized phase. This is the only time in the search that the value function, $\tilde{V}^{\pi}_{H'-1}$, is taken into account. We base this phase on a *first-best hill climber* using the evaluation function,

$$z = C(s, \mathbf{a}) + \frac{1}{N} \sum_{j=1}^{N} \tilde{V}^{\pi}_{H'-1}(f(s, \mathbf{a}, \omega^j)) \tag{6.12}$$

based on the expression in (6.7). Our implementation of GRASP for the problem of searching for a suitable action $\mathbf{a} \in A_s$ is presented in Algorithm 13.

Here, we construct the neighborhood, $\mathcal{N}$, from an enumeration of every feasible single move of a procedure to a new room or day along with all feasible swaps between two procedures. We terminate the local search procedure by using an upper bound on evaluations without improvement, or if the entire neighborhood has been evaluated.

To make all solutions to the action $\mathbf{a}$ comparable, the disturbances that are required for the SAMW algorithm, as well as for (6.12), are generated during the initialization of the algorithm. Furthermore, we reuse the $\mathcal{T}$ sequences, $\omega^j_2, \omega^j_3, \ldots, \omega^j_{H'}$ between each sample path j. So, accounting for the model horizon, H', the number of iterations in the SAMW algorithm, $\mathcal{T}$, and the N subsequent states, we require a total of $N + \mathcal{T} \cdot (H' - 1)$ randomly generated disturbances for the execution of Algorithm 13.

6.4 Implementation & Results

In this section, we demonstrate our simulation-based MDP based on data from a Danish hospital. We use data on patient arrivals and ward resources to estimate the occurrence of requests, procedure duration, and room availability.

In Section 6.4.1 we present the hospital case along with a number of assumptions related to our model implementation. Next, in Section 6.4.2 we present the parameter tuning, followed by Section 6.4.3 where our approach is compared to a range of myopic policies using simulation.

Algorithm 13 Our GRASP implementation in the search for an action $\mathbf{a} \in A_s$

$\mathbf{a}^* \leftarrow (0, 0, \cdots, 0)^T$
$z^* \leftarrow \infty$
$disturbances \leftarrow generate()$ ▷ Generate all required disturbances: ω_t^j
while $stillTimeLeft$ **do**
    $\mathbf{a} \leftarrow (0, 0, \cdots, 0)^T$
    for all $\sum_{i,j \in P,J} p_{ij}$ **do**
        $list \leftarrow getCandList(\mathbf{a}, s, \alpha)$
        $\mathbf{a} \leftarrow pickRandom(\mathbf{a}, list)$ ▷ Pick randomly from restricted candidate list
    end for
    $y \leftarrow 0$
    $\mathcal{N} \leftarrow getNeighborhood(\mathbf{a}, s)$
    while $y < noImpromentent$ and $y < |\mathcal{N}|$ and $stillTimeLeft$ **do**
        $z, i \leftarrow evaluateNewRandom(\mathcal{N}, disturbances)$ ▷ Evaluate random element i from $\mathcal{N}$
        if $z < z^*$ **then**
            $z^* \leftarrow z$
            $\mathbf{a}^* \leftarrow update(\mathcal{N}, i)$
            $\mathcal{N} \leftarrow getNeighborhood(\mathbf{a}^*, s)$
            $y \leftarrow 0$
        else
            $y \leftarrow y + 1$
        end if
    end while
end while
return $\mathbf{a}^*$

Type	Occurrences pr. day	Duration Mean (h)	Duration Variance (h^2)
Procedure A	0.57	2.26	2.15
Procedure B	0.49	2.26	2.05
Procedure C	0.14	1.44	1.41
Procedure D	0.11	2.91	2.99
Procedure E	0.10	1.25	1.01
Procedure F	0.07	1.64	1.25
Procedure G	0.07	1.75	1.37
Procedure H	0.06	2.69	2.30
Procedure I	0.06	2.56	2.12
Procedure J	0.06	1.65	1.40

Table 6.1: Occurrence rate, sample mean duration, and variance obtained for the ten most frequent types of requests. These account for about 52% of the total occurrence rate.

6.4.1 Case & Data Description

For our hospital case, requests occur according to $|P| = \mathbf{288}$ different types. The occurrence-process is further assumed to be stationary and Poisson with known parameters. Each request will be subject to an availability-pattern for which we assume that every successive period of five days has *at most* one day where the designated physician is unavailable. In addition, all patterns occur with equal probability. Furthermore, the procedure duration is random, but with known mean and variance.

Data for the ten most frequent types of requests, accounting for 52% of the total occurrence rate, are presented in Table 6.1.

We assume that all requests have to be allocated. However, if the hospital does not have sufficient capacity within the current planning horizon, then a minimum number of requests are allowed to be outsourced. The fixed planning horizon is set to $H = \mathbf{20}$ days within which the capacity on the number of open rooms depends on the weekday, as shown in Table 6.2. In total, the hospital has three different rooms at disposal for which the opening-hours results in a total time-capacity of $w_k = \mathbf{7.5}$ hours.

The planner further has to account for equipment compatibility between procedure types and rooms. The compatibility between procedures and rooms for the ten most frequent types are presented in Table 6.3, where 1 indicates that the procedure is compatible with the room; otherwise the indicator is 0. Between all allocated procedures we assume a fixed buffer time of $m = \mathbf{0.5}$ hours.

Weekday	Monday	Tuesday	Wednesday	Thursday	Friday
Limit on open rooms	1	2	2	2	2

Table 6.2: Upper limit on number of open operating rooms for each respective weekday.

Type	Room A	Room B	Room C
Procedure A	1	0	0
Procedure B	1	1	0
Procedure C	1	1	1
Procedure D	1	1	1
Procedure E	1	1	1
Procedure F	1	1	1
Procedure G	1	1	1
Procedure H	1	1	1
Procedure I	1	1	1
Procedure J	1	1	1

Table 6.3: Compatibility between procedure types and rooms. Shown for the ten most frequent types. The number 1 indicates the a procedure is compatible with a room; otherwise the indicator will be 0.

Model Implementation

We assume that the capacity utilization of room $k \in R$ is distributed according to a log-normal distribution with probability density function

$$p_k(\tau) = \frac{1}{\varrho_k \tau \sqrt{2\pi}} \cdot e^{\frac{-(\ln \tau - \gamma_k)^2}{2\varrho_k^2}} \tag{6.13}$$

where $\gamma_k = \ln(\mu_k^2/\sqrt{\sigma_k + \mu_k^2})$, $\varrho_k = \sqrt{\ln(\sigma_k/\mu_k^2 + 1)}$, μ_k is the sum of the expected durations for all procedures allocated to room $k \in R$, and σ_k^2 is the corresponding sum of their variances. In addition, we use a polynomial function to evaluate the cost of performing procedures in overtime δ, assuming that $f(0) = 0$. That is,

$$f(\delta) = b_1\delta^2 + b_2\delta \tag{6.14}$$

In practice the parameters b_1 and b_2 would be adjusted to attain the desired slope and relation to the payed overtime-costs, and the more intangible costs of stretching the procedure duration into overtime. Later, we will assess the result of adjusting these parameters on the performance of our model.

For the SAMW algorithm we employ two base-policies in the set Λ. These have been chosen to account for the uncertainties in the resulting costs and at the same time maintain a reasonably fast evaluation time. We will refer to these base-policies as:

1. The **Anticipative Increased Cost Policy** (AIP)
2. The **Anticipative Weighted Cost Policy** (AWP)

In both policies, the current requests, p_{ij}, are allocated to the schedule, r_{ikl}, according to their expected duration, $E[Y_i]$, in ascending order. Each request at a time, they evaluate all *feasible* room-day pairs, $k, l \in R, T$, within

the planning horizon and allocate the requests based on the lowest *anticipative* cost. The latter is estimated differently in each of the policies.

1. The AIP estimates the increased cost similar to (6.11), but for the difference, $o^i_{kl} - o^{i-1}_{kl}$, accounts for the future procedures that have not appeared in the schedule, yet. Specifically, the total capacity utilization is estimated from μ_{kl} and σ_{kl} (cf. the distribution in (6.13)), where each parameter is a sum of the already allocated procedures and an estimate of the future procedures. Thus, prior to allocating the request, the AIP assumes that

$$\mu_{kl} = \eta_l \cdot E[Y_G] + \sum_{i \in P}(r_{ikl} \cdot E[Y_i]) + (\sum_{i \in P}(r_{ikl}) - 1) \cdot m \tag{6.15}$$

and

$$\sigma^2_{kl} = \eta_l \cdot Var(Y_G) + \sum_{i \in P}(r_{ikl} \cdot Var(Y_i)) \tag{6.16}$$

for each feasible room-day pair, $k, l \in R, T$, where $E[Y_G] = \sum_{i \in P} E[Y_i] \cdot \lambda_i/\lambda_G$ is the global weighted average duration, $Var(Y_G) = \sum_{i \in P} Var(Y_i) \cdot \lambda_i/\lambda_G$ is the global weighted average variance, and $\lambda_G = \sum_{i \in P} \lambda_i$ is the global rate of occurrence. Lastly, $\eta_l \in \mathbb{R}_{>0}$ estimates the additional number of requests that day $l \in T$ will be subject to in the future. Thus,

$$\eta_l = \sum_{x=t+1}^{l} (\lambda_G/(H \cdot |R| - d_x)) \tag{6.17}$$

for $l \geq t + 1$; otherwise $\eta_l = 0$. Further, $d_x \in \mathbb{N}_0$ is the number of room-day pairs that are closed (due to capacity depletion) within the horizon relative to day $x \in \mathbb{N}_{t+1 \leq x \leq l}$.

2. For the AWP policy, each allocation depends again on the requests that have not appeared in the schedule yet. However, the AWP is based on the notion that uncertainty should be employed as a "weight" rather than an estimate of the potential overtime-costs. Consider the difference $o^i_{kl} - o^{i-1}_{kl}$ from (6.11). This time o^{i-1}_{kl} is evaluated by merely summing over the known procedures in r_{ikl}, whereas the resulting overtime-cost, o^i_{kl}, is based on (6.15)-(6.17). However, we change (6.17) to $\eta_l = \nu \cdot \sum_{x=t+1}^{l}(\lambda_G/(H \cdot |R| - d_x))$, where $\nu \in \mathbb{R}_{>0}$ determines the "weight" of these uncertain requests, and is determined experimentally.

6.4.2 Adjusting the Parameters

The MDP model parameters were assessed and adjusted by applying the model to a simulation framework. That is, we simulated the arrival of requests

and their resulting utilization of capacity in the system by generating pseduo-random numbers. In this simulation, we have assumed that requests occur according to a Poisson process, and that the total capacity utilization of any room is distributed according to a log-normal distribution as defined by (6.13).

We randomly generated three different sets of seeds covering a simulation period of 565 days, and then replicated each run of the simulation on each respective set twice. 365 days were used to burn-in the simulation for which we used the AIP policy to save runtime, leaving 200 days to assess the model performance of the MDP. We employed a fixed runtime of 20 minutes, and conducted tests with three different levels for each respective parameter. The parameters that were subject for testing, and their levels, are presented in Table 6.4. The number of sampled paths, N, and the SAMW iterations, $\mathcal{T}$, were tested with interaction resulting in a total of $(3 \times 3 + 3 + 3) \times 2 \times 3 = 90$ simulations. The remaining parameters were adjusted during preliminary testing of the model.

For the cooling schedule, β_i, we tested both a fixed cooling parameter, such that β_i remained constant for all $\mathcal{T}$ iterations, and an exponential decreasing continuous function, $\beta_i = \beta(i) = 1 + C(\epsilon, \mathcal{T})^{-(i-1)}$, where $C(\epsilon, \mathcal{T}) = e^{-\frac{ln(-1+\epsilon)}{\mathcal{T}}}$, and ϵ defines the final cooling value after $\mathcal{T}$ iterations.

Lastly, for the overtime function in (6.3) we used a setting with $b_1 = 10$ and $b_2 = 4$, and a fixed setup-cost of $\kappa = 100$. The penalty for outsourcing requests was set to $\phi = 1 \cdot 10^6$.

#	Parameter	Level 1	2	3
1	SAMW Iterations ($\mathcal{T}$)	$\mathbf{20}^*$	50	100
2	Cooling Schedule	$\beta_i = 2$ (fixed)*	$\epsilon = \mathbf{1.1}$	$\epsilon = 1.001$
3	Sampled Paths (N)	$\mathbf{5}^*$	10	30
4	Model Horizon (H')	50	$\mathbf{150}^*$	300

Table 6.4: Parameters that have been subject to simulation. Parameter 1 & 2 are related to the SAMW algorithm (Algorithm 11), and 3 & 4 to the rollout expression (6.7). The bold font indicates the preliminary setting of the parameters during the simulations, whereas the star indicates the setting employed in the experiments in Section 6.4.3.

We measured the performance of each setting on the cumulated cost (over the 200 day simulation period) of both the overtime- and setup-costs; and the penalty from outsourcing requests. The parameters were then compared in a dot-plot and on their respective correlations to the amount of cumulated value. Interestingly, the cooling schedule showed to be more effective when held constant at $\beta_i = 2$ and decreasing in performance as ϵ increases; hence when β_i decreases at a faster rate. The cooling schedule had a distinct effect on the performance, whereas the remaining parameters were more inconclusive. The effect from the number of SAMW iterations, $\mathcal{T}$, was almost negligible with respect to the cumulated value, whereas the number of sampled paths, N, and the model horizon, H', depended more on the specific set of seeds for

the simulation.

As regards the value of ν in the AWP, we employed a hill climber heuristic where the average performance was recursively evaluated over ten different sets of seeds until convergence. This resulted in a final weight of $\nu = 16.969$.

6.4.3 Numerical Experiments

In this section, we apply our MDP model based on the results from the parameter tests, and compare the performance to a range of different policies. These include a policy that resembles the behavior of a "manual" planner, which we will refer to as the Manual Policy (MP). Next, we compare the MDP performance to a more advanced heuristic search procedure.

The MP is based on the following assumptions:

1. The expected duration of each procedure is known to the planner.

2. The planner is familiar with procedure variability, but the exact distribution nor spreading is not known. For this reason, a fraction of the available capacity is used as a buffer such that a new procedure is not allowed to start within this time-interval. However, the total expected duration of the allocated procedures is allowed to violate the buffer capacity by at most 10% of the total capacity.

3. The exact costs are unknown to the planner. For this reason, the planner will try to utilize the setup-cost for a new room as much as possible. Firstly, the requests are sorted in ascending order similar to the policies in Λ, and then allocated in sequence to the room-day pair that results in the least amount of excess capacity. If there are no feasible allocations for the room-day pairs that are already in use, the planner will allocate the request to the latest unused room-day pair such that this new room will be subject to as many future requests as possible.

Our experiments were conducted using simulations similar to the tests in Section 6.4.2. Thus, a period of 365 days were used to burn-in the simulation, and 200 days to assess the performance of the model. However, simulations were extended to eight different sets of seeds and replicated five times on each set. Besides testing the model on a range of different seeds, we varied the parameters in the overtime function (6.14) on four different levels, presented in Table 6.5. Later, we will refer to the overtime-cost settings using the conventions presented in this table. Again, the MDP runtime was fixed at 20 minutes for each day over the entire length of the simulation.

Our experiments in this section include the MDP, MP with a capacity buffer of both 10% and 20%, and the policies in Λ.

Reference	b_1	b_2
Low	10	4
Medium	100	10
High	300	100
Very High	10,000	500

Table 6.5: Parameter settings for the overtime function (6.14). All four levels are tested at each of the eight sets of seeds.

In order to compare the performance across the different combinations of seeds (and thereby the behavior of the requests generated) and overtime-costs, we standardized the cumulated cost, including the penalty for outsourcing, by employing the conversion

$$x_{ijk} = \frac{y_{ijk} - \min_{k \in \mathcal{K}_{ij}}\{y_{ijk}\}}{\max_{k \in \mathcal{K}_{ij}}\{y_{ijk}\} - \min_{k \in \mathcal{K}_{ij}}\{y_{ijk}\}} \tag{6.18}$$

where y_{ijk} is the resulting cost of simulation run $k \in \mathcal{K}_{ij}$ using seeds i and overtime-cost setting j. Thus, for the five replications of the MDP, the MP with both 10% and 20% capacity buffer; and the policies in Λ, $|\mathcal{K}_{ij}|$ includes 9 runs for each combination of i and j.

The results are presented in Table 6.6 showing the performance for each model and overtime-cost setting, presented as both the average and standard deviation standardized cost. The table shows a distinct difference between the MDP and the remaining policies, measured on both the average and standard deviation performance. The difference is especially distinct between the MDP and the MP, regardless of the capacity buffer. Notice that the MP with 20% capacity buffer yields an average of 1.000 and a standard deviation of 0.000 for the first three overtime settings because this policy resulted in the highest cost across all eight sets of seeds.

Interestingly, the benefit of using the MDP increases as function of the overtime cost. Simultaneously, the difference between the policies decreases, resulting in quite indifferent performance at the highest overtime cost level. Otherwise, the MP performs substantially better with a 10% instead of a 20% capacity buffer. Still, the anticipative policies in Λ yield substantially lower costs than both MP settings, where AWP results in both lower average and standard deviation cost than all the remaining policies, except at the highest level. The relative difference between the average performance of the MDP and AWP is quite large, but given an average runtime of about 3 milliseconds for the AWP, the latter might still be a suitable choice in many operational settings.

	Average					Std. Deviation				
Overtime	MDP	MP 10%	MP 20%	AIP	AWP	MDP	MP 10%	MP 20%	AIP	AWP
Low	0.009	0.473	1.000	0.188	0.075	0.014	0.120	0.000	0.149	0.053
Medium	0.075	0.499	1.000	0.234	0.096	0.027	0.130	0.000	0.161	0.050
High	0.015	0.526	1.000	0.267	0.136	0.016	0.133	0.000	0.158	0.054
Very High	0.019	0.977	0.927	0.800	0.857	0.016	0.037	0.097	0.107	0.070

Table 6.6: Performance of the MDP compared to the MP with a capacity buffer of 10% and 20%; and the base-polices in Λ. The models are compared on their standardized cost.

We should emphasize that the performance of the MDP, AIP and AWP are only relevant in practice if the necessary computational setup can be introduced into the hospital operations, as is an obvious advantage of a more simple policy — such as the MP. However, if this *is* the case, then we should consider how other computational methods compare to the MDP performance, which will be elaborated in the following section.

Further Validation

In this section, we compare our MDP to a GRASP heuristic with a myopic structure. That is, we re-used the basic algorithmic structure that was presented in Algorithm 12, but without the anticipative costs. Instead, during the local search, we evaluate the solution on the sum of the expected overtime-cost and the fixed setup-cost over the entire planning horizon. Thus,

$$C'(s,\mathbf{a}) = \sum_{k,l\in R,T\setminus\{t+H\}} o_{kl} + \sum_{k,l\in R,T\setminus\{t+H\}} (y^{s^*}_{kl} - y^{s}_{k,l+1})\cdot\kappa + \sum_{i,j\in P,J} q_{ij}\cdot\phi \quad (6.19)$$

Just as in our previous experiments, we simulated the performance of the GRASP heuristic with 20 minutes of runtime, and replicated each run five times on each combination of seeds and overtime-cost setting.

The result of the simulations are presented in Table 6.7, showing the average and standard deviation performance for each model and overtime-cost setting. The average performance has further been depicted in Figure 6.3. Again, the models are compared on their standardized cost according to (6.18), but re-calculated to fit the only two models that are compared in this section.

	Average		Std. Deviation	
Overtime	MDP	GRASP	MDP	GRASP
Low	0.276	0.201	0.422	0.354
Medium	0.302	0.357	0.328	0.382
High	0.392	0.463	0.365	0.411
Very High	0.453	0.493	0.294	0.303

Table 6.7: Performance of the MDP compared to the GRASP heuristic. The models are compared on their standardized cost

Table 6.7 shows that the performance of the two models is much more equal compared to our previous experiments. In fact, the GRASP outperforms the MDP in both average and standard deviation cost when the overtime-cost is set to "Low". This corresponds to the parameters $b_1 = 10$ and $b_2 = 4$ which is the setting that the MDP was adjusted for. However, as the cost of stretching procedures into overtime increases, so does the MDP performance resulting in lower average (cf. Figure 6.3) and standard deviation cost for the remaining overtime settings.

We should further emphasize that in the cheapest setting, the overtime-cost does not exceed the cost of opening a new room until about 3 hours into overtime, which is longer than the expected duration for most of the occurring requests in our data. This may not apply to many real hospital settings. In addition, these experiments were conducted for a reasonably short simulation of 200 days; hence, if the improvement of exploiting the rolling and overlapping nature of the problem is small, then such advantage might only show for much longer periods of simulation.

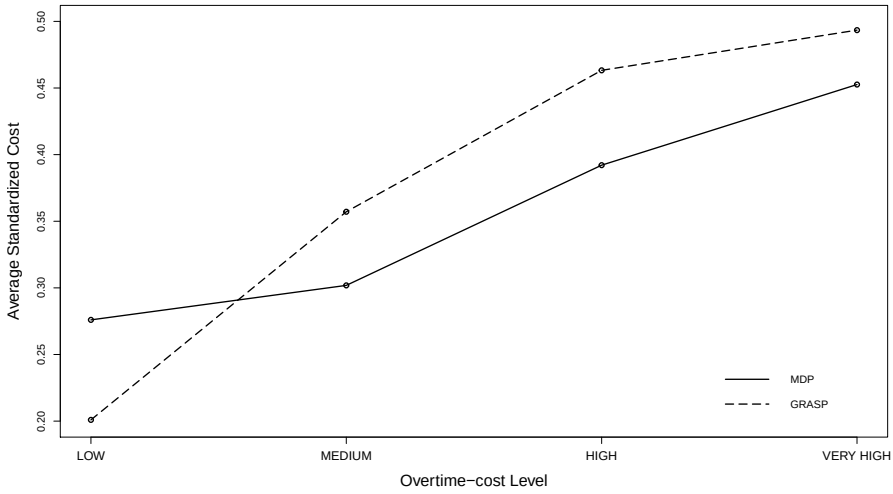


Figure 6.3: Performance of the MDP and GRASP measured on their standardized cost. On average, the MDP yields better decisions when the overtime-cost is anything but at the cheapest level.

6.5 Conclusion

Aiming to apply and test a new approach to the problem of scheduling operating theatres, we developed a simulation-based Markov Decision Process (MDP). The advantage of such modeling approach is that a sequence of decision problems are taken into account, which seems to have been disregarded

in this area of scheduling in general. In fact, to our knowledge the MDP approach has not been considered previously in day-to-day scheduling of requests to operating theatres.

Specifically, our approach consists of deriving a heuristic rollout policy, evaluating each action of the current state, based on a sampling of a number of potential future paths. This process is further based on a predefined set of base-policies by employing an algorithm known as Simulated Annealing Multiplicative Weights (SAMW) [36]. We further consider that the state-dependent action space is intractable, and for this reason we derive an action by employing a Greedy Randomized Adaptive Search Procedure (GRASP).

In order to validate our MDP, we conducted a number of numerical experiments based on simulation, where we compared our approach to both simple and more advanced myopic scheduling methods. Firstly, we validated a policy that resembles a manual planner, which indicated that there is a *distinct* improvement of employing our model rather than scheduling requests manually. Furthermore, we found that a substantial improvement can be attained by employing a policy that accounts for future requests by weighting their contribution to the overtime-costs. We refer to this as the Anticipative Weighted Cost Policy (AWP). In addition, we found that a GRASP disregarding the rolling horizon performs only slightly worse than our MDP, and in fact better when the cost of stretching into overtime is sufficiently low. In other words, the myopic GRASP might be beneficial to certain hospital cases. However, consider that better results might be achievable for other base-policies in Λ and more effective implementations of the SAMW algorithm.

6.5.1 Future Work

In future work more extensive numerical experiments should be considered. The difference in performance between the simulation-based MDP and myopic GRASP should be investigated by extending the period over which simulation is conducted, employing more levels on the overtime-cost setting, and more effective base-policies in Λ. Additionally, further work into more simple policies should be investigated to benefit the hospital cases where requests have to be allocated within a short time (e.g. below a few seconds).

Acknowledgments

This research was supported by the Danish governmental organization Region Sjælland. In particular, we would like to thank the department of Production, Research and Innovation for providing us with essential data and information on the operations of the Danish hospital operating theatres. In addition, we would like to thank Assistant Prof. Charlotte Vilhelmsen for providing us with insight into the literature on scheduling operating theatres and Professor Bo Friis Nielsen with methodological advice.

Part IV

Conclusion

Chapter 7

Conclusion, Perspective & Future Work

7.1 Conclusion

One of our main objectives has been to provide management and planners with a range of tools that can be employed to improve the utilization of hospital resources related to patient flow. In achieving this, we have given particular attention to inpatient flow, which constitute one of the core operations of a hospital (cf. Section 1.2). We have provided a method for modeling the occupancy of the hospital wards, and methods for optimizing the configuration of resources related to both room types and the aggregated bed capacity. In addition to inpatient flow, we have presented an approach for modeling an emergency department with time-dependent behavior, and a method for optimizing the associated resources. Lastly, we have investigated an approach for optimizing an area of the hospital operations that interact with several patient flow types, namely the process of scheduling surgical procedures.

All things considered, we have covered a wide range of problems related to hospital patient flow.

In relation to expanding the current knowledge of modeling and optimizing patient flow, we have mainly been focusing on methods within two different fields, namely Markov chain modeling and heuristic optimization.

Firstly, we have aimed at expanding the scarce amount of Markov chain models that have been used to model patient flow as a system of queues [22]. Thus, throughout this thesis we have demonstrated how to model and apply continuous-time Markov chains to different relevant hospital problems.

Secondly, we have employed these Markov chain models in a number of heuristic search procedures, which in the literature is a rather uncommon approach. In fact, for the context of optimizing patient flow, our literature reviews indicate that only few studies integrate both optimization *and* analytical stochasticity in patient flow. Often a number of pre-defined scenarios are evaluated, or for the instances where optimization is employed, the patient flow is evaluated using simulation. As regards scheduling of surgical patients, a substantial amount of studies employ uncertain requests (e.g. acute patients), but only a few of these consider the inherent rolling horizon of the problem.

As a result of the aforementioned, this thesis has lead to a number of both

heuristic and matheuristic search procedures that are able to exploit the behavior of patient flow to provide useful solutions for both hospital management and planners.

7.1.1 Specific Findings

In the following we will elaborate on the specific findings of Chapter 3-6.

In **Chapter 3**, we set out to provide hospital management with a decision tool for improving the utilization of bed resources. Specifically, we considered a set of patient types arriving to a set of inpatient wards, and additionally that patients can be relocated to alternative wards whenever the ward capacity is insufficient. We further considered a situation where a hospital is not able to expand their total bed capacity, and must optimize the utilization of their current resources by minimizing the expected number of patients that are relocated on arrival.

To achieve this, we modeled the flow of patients as a queueing system based on a homogeneous continuous-time Markov chain. Our model accounts for N wards and N different patients, as well as the arrival and service rates for each patient type. Furthermore, if a relocation is needed, our model accounts for the probability of routing a patient of a specific type to a specific ward in the system. Lastly, in aiming to make this Markov chain model applicable in a practical setting, we provided a method for minimizing the computational resources that are needed to evaluate the steady-state probability distribution by truncating the state space. We validated this modeling approach by employing data from a Danish hospital in a heuristic statistical test, which showed that our model adequately reflects the bed occupancy of a real hospital setting.

In order to minimize the expected number of patient relocations, we employed our Markov chain model in the objective function of a heuristic search procedure with a hill climber structure. Based on data from a real hospital case, our search procedure was able to reduce the objective value by 11.8% compared to the hospital's current distribution of beds. A complete enumeration of the problem showed that this was in fact the optimal solution to the problem.

In **Chapter 4**, we extended the optimization problem considered in Chapter 3 by including the configuration of room types among the wards. Instead of minimizing the expected number of relocated patients, we used an objective function that maximizes the expected number of patient preference-matches for private rooms. The amount of relocations were instead included as a constraint in the optimization problem. Similar to Chapter 3, we considered a situation where a hospital is subject to a fixed number of each respective room type, and must improve their service by changing the configuration of these rooms.

We achieved this by employing a heuristic search procedure that samples solutions from the search space based on an interpolation between the cur-

rently known solutions. In this process, the objective value of each solution is evaluated by deriving the occupancy distributions for patients that prefer admission to private rooms from the aggregated occupancy distributions. The latter is calculated by employing the Markov chain model from Chapter 3. Furthermore, in the initialization phase of this search procedure, uniform samples are obtained from the search space by using a fast surrogate objective function. This sets the stage for the solutions that are sampled and evaluated with the true, but much slower, objective function.

Based on data from both a Danish and Belgian hospital, we validated the performance and robustness of our search procedure, and found that we were able to derive near-optimal solutions within a relative gap of 1% from the optimum. To further validate our approach, we conducted simulations which showed that the obtained room configurations benefit the day-to-day process of scheduling patients to rooms.

In **Chapter 5** we aimed at deriving a decision support tool for hospitals that are generally governed by their efficiency, and therefore seek to rearrange their resources by analyzing the difference between the currently available and minimum required department resources. More specifically, our focus was to minimize the amount of staff resources for an emergency department and simultaneously account for constraints on the patient waiting times.

Similar to the aforementioned studies, we modeled the patients moving through the emergency department by using a continuous-time Markov chain model. However, since the arrival rate and the availability of staff is distinctly fluctuating for this type of patient flow, we introduced time-dependent behavior into the queueing system. This was achieved by employing a piecewise transient model, where the state probability distribution is recursively evaluated with uniformization. We validated this modeling approach by comparing to various simulations of the associated system. In these experiments, we found that our approach adequately can model patient waiting times in a time-dependent system, and is reasonably robust to different service-time distributions.

In order to minimize the emergency department's staffing, we employed our Markov chain approach in a matheuristic search procedure that recursively adapts and solves an integer linear programming model based on evaluations of the patient waiting times. The resulting solution is then passed to a tabu search heuristic that further minimizes the amount of staff. We tested this approach on a number of different input datasets, and found that tabu search was only able to improve the solution in a single case.

We validated our optimized staff configurations in simulations with multiple triage-classes of patients, and found that our solutions performed well with only slight violations for the least prioritized patients. These results were despite that we only accounted for a single class when the solutions were derived.

In **Chapter 6** we set out to provide hospital planners with an operational decision tool for scheduling surgical patients, resulting in a minimization of the

total expected long-term costs related to this process. We considered a situation where requests for a procedure occur continually, but are not scheduled until the end of the day. At this point, the planner must account for the potential overtime-costs, setup costs, and a range of constraints, which among other things relate to the planning horizon and capacity of the system.

Aiming to achieve this, we investigated an approach that accounts for the inherent rolling horizon of the problem, and thus makes the scheduling process anticipative. Specifically, we modeled the optimization problem as a simulation-based Markov decision process. That is, we employed a heuristic "online" approach, referred to as rollout, where an action is derived based on an evaluation of potential future paths with a simulated annealing multiplicative weights algorithm. Moreover, we avoid enumerating the entire action space by employing a Greedy Randomized Adaptive Search Procedure (GRASP) such that only the most promising candidates of the action space are evaluated.

We validated our model by simulating the performance with different input data, and compared the results to a range of other heuristic approaches. In these experiments, our model exercised distinct improved performance over the simple policies, and especially a policy that resembles a manual planner. Additionally, we found that a substantial improvement of the scheduling process can be attained by employing a policy, which we refer to as the anticipative weighted cost policy.

Further experiments showed that the performance of our model depends on the cost of conducting the surgical procedures in overtime, and that a myopic GRASP only performs slightly worse, despite that the rolling horizon was omitted.

7.2 Perspective & Future Work

Even though hospital planning is a popular research field, we have found that there is still a wide range of opportunities when it comes to understanding and improving the processes that govern patient flow. We have shown that Markov chain modeling can be a viable basis for a number of relevant hospital optimization problems, and that accounting for a rolling horizon may benefit the problem of scheduling surgical patients. For this reason, we hope that our results can serve as the basis for further research in the field of hospital patient flow.

Even so, we acknowledge that our methods yield a couple of obstacles that cannot be ignored. A noteworthy obstacle that applies to all four studies in this thesis, is that real-life systems often lead to very large and computationally expensive state spaces. This is especially problematic in the context of optimization, where many successive runs need to be conducted in order to derive a good solution. To cope with this problem, Chapter 3-5 relied on truncating the state space, whereas the system in Chapter 6 was evaluated with simulation, and thus excluded the analytical approach. For the Markov chain

models that were truncated, we deem that certain parameters can still cause the state space to become intractable, at least if the purpose is optimization and the model accuracy has to be maintained. For instance, the state space in Chapter 3-4 increases as function of the ward number, and the model must therefore rely on groups of patients with similar length of stay to confine the amount of states. For the state space in Chapter 5, the occupancy of patients in the system depends on the balance between arrival rate and the emergency department staffing.

A related issue is coping with the arrival process and the service time distribution of the system. In this thesis, we have found that a Poisson process is an adequate assumption for inpatients, and shown that acute arrivals are governed by a Poisson process with a weekly cyclical pattern. Regarding the service time distribution, we have assumed that this is exponential throughout Chapter 3-5 for convenience. Our experiments showed that the queueing systems considered in this thesis are generally robust to this assumption, and that certain inpatient groups are actually governed by distributions that are close to exponential. Thus, we deem that this assumption may well be reasonable for a much wider variety of cases, than we have considered here.

Turning to the perspective of optimization, the reader may have noticed that our methods require rather long runtimes, which make them more useful to strategic and tactical planning, and less to contexts where a decision is required within few minutes. Even so, we should emphasize that for the cases where we have employed an analytical model of the patient flow, the excessive calculations are accompanied by a high accuracy, an opportunity for other researchers to reproduce the results, and a basis for obtaining bounds, which might lead to solving these optimization problems to proven optimality at some point in the future.

Lastly, but certainly not least, our methods have been focused on the specific application area of hospital patient flow. Nonetheless, there are a wide range of similar application areas for which we deem that our findings are just as applicable. Materials flowing through or between factories have characteristics that are similar to patient flow. For instance, factory management may wish to optimize the distribution of tools among machines, or determine when and where to process certain jobs. The same applies to call centers that have to schedule operators, and minimize their costs by simultaneously accounting for the customer waiting time. Furthermore, the performance of certain computer systems can be improved by considering the relation between randomly occurring jobs and the processor configuration.

7.2.1 Future Work

In this thesis, we have opened the door to a variety of future research projects, presented below:

Regarding the distribution of ward resources for inpatients (cf. Chapter 3

and 4), further research into increasing the number of wards should be considered. In this thesis, we have only accounted for a subset of the wards that constitute an entire hospital, whereas including all wards might lead to better model fits, as well as an improved distribution of resources. Further, as the number of wards increases, more constraints may appear in the optimization problem, and thus the structure of the current search procedures may have to be redefined. Finally, simulation experiments with different input (e.g. service time distributions) can be useful in deriving the conditions under which our modeling approach is no longer adequate.

For the scheduling of room resources specifically, more complex cases should be investigated, for instance by introducing gender types, and that the fraction of both room preferences and gender can be a function of the preferred ward.

For the optimization of acute patient flow (cf. Chapter 5) research should be conducted to improve the search procedure, for instance by testing different approaches to adjusting the bound on staffing. Even more important, in order to properly assess the solutions and perhaps even determine that the optimal solution has been found, future research should investigate whether lower bounds on the optimal objective value can be obtained. Any research in this direction, may eventually lead to algorithms that can guarantee proven optimality, which will be highly valuable. This point is relevant to the aforementioned studies on inpatient flow as well.

In addition, to ensure that our model fits the behavior of acute patient flow, further data on the emergency department should be uncovered to properly assess our modeling approach. That is, the actual service time distributions, the occupancy of patients at each node in the system, and the distribution of patient waiting time.

For the scheduling of surgical patients (cf. Chapter 6), further simulation experiments should be conducted to clarify the difference in performance between our simulation-based Markov decision process and the myopic GRASP. From here, the results could further be compared to the optimal myopic solution.

Experiments with different base-policies should be considered with a view to improve our approach. As we have already seen, such base-polices can be employed as stand-alone scheduling approaches that may be able to outmatch the more advanced methods in many practical settings due to their fast runtimes.

In general, a greater number of simulation experiments, or analytical work, would help to clarify the conditions under which the (simulation-based) Markov decision process is a practical approach to the scheduling problem. For instance, research on lower bounds might reveal when the potential is small, and that a different model is more beneficial. Additionally, experiments with more complex systems may show that a myopic approach (which is often able to capture more system characteristics) usually obtains better solutions.

Bibliography

[1] Arena simulation software - rockwell automation. https://www.arenasimulation.com/. Accessed: 07-06-2018.

[2] Flere fejl på afdelinger med overbelægning. http://www.danskepatienter.dk/nyheder/flere-fejl-p-overbelagte-afdelinger. Accessed: 18-03-2016.

[3] The life of a.k. erlang. https://web.archive.org/web/20061203043554/http://oldwww.com.dtu.dk:80/teletraffic/erlangbook/pps009-022.pdf. Accessed: 19-04-2018.

[4] Overbelægning er stadig et problem på sygehusene. http://www.danskepatienter.dk/nyheder/overbel-gning-er-stadig-problem-p-sygehusene. Accessed: 18-03-2016.

[5] Plant simulation - siemens. https://www.plm.automation.siemens.com/fr/products/tecnomatix/manufacturing-simulation/material-flow/plant-simulation.shtml. Accessed: 07-06-2018.

[6] Region sjælland. http://www.regionsjaelland.dk/Kampagner/English/Sider/default.aspx. Accessed: 28-05-2018.

[7] Sikkert patientflow – erfaringer fra et forbedringsprojekt. Dansk Selskab for Patientsikkerhed, 2015.

[8] Shola Adeyemi, Thierry Chaussalet, and Eren Demir. Nonproportional random effects modelling of a neonatal unit operational patient pathways. *Statistical Methods and Applications*, 20(4):507–518, 2011.

[9] Shola Adeyemi and Thierry J. Chaussalet. A random effects sensitivity analysis for patient pathways model. *Ieee Symposium on Computer-based Medical Systems (cbms)*, pages 536–538, 2008.

[10] Mohamed A. Ahmed and Talal M. Alkhamis. Simulation optimization for an emergency department healthcare unit in kuwait. *European Journal of Operational Research*, 198(3):936–942, 2009.

[11] Renzo Akkerman and Marrig Knip. Reallocation of beds to reduce waiting time for cardiac surgery. *Health Care Management Science*, 2009.

[12] Theodore T. Allen. *Introduction to Discrete Event Simulation and Agent-based Modeling*. Springer London, 2011.

[13] Anders Reenberg Andersen, Bo Friis Nielsen, and Line Blander Reinhardt. Optimization of hospital ward resources with patient relocation using markov chain modeling. *European Journal of Operational Research*, 260(1):1152–1163, 2017.

[14] R. Andriansyah, T. Van Woensel, F. R. B. Cruz, and L. Duczmal. Performance optimization of open zero-buffer multi-server queueing networks. *Computers and Operations Research*, 37(8):1472–1487, 2010.

[15] Mor Armony, Shlomo Israelit, Avishai Mandelbaum, Yariv N. Marmor, Yulia Tseytlin, and Galit B. Yom-Tov. On patient flow in hospitals: A data-based queueing-science perspective. *Stochastic Systems*, 5(1):146–194, 2015.

[16] Rym Ben Bachouch, Alain Guinet, and Sonia Hajri-Gabouj. An integer linear model for hospital bed planning. *International Journal of Production Economics*, 140(2):833–843, December 2012.

[17] Norman T. J. Bailey. A study of queues and appointment systems in hospital out-patient departments, with special reference to waiting-times. *Journal of the Royal Statistical Society. Series B (methodological)*, 14(2):185–199, 1952.

[18] Jørgen P. Bansler and Jes Søgaard. Så lyt dog til kritikken af sundhedsplatformen. *Politiken*, pages 7–8, 2017.

[19] Nicola Bartolomeo, Paolo Trerotoli, Annamaria Moretti, and Gabriella Serio. A markov model to evaluate hospital readmission. *Bmc Medical Research Methodology*, 8(1):23, 2008.

[20] S. Batun, B.T. Denton, T.R. Huschka, and A.J. Schaefer. Operating room pooling and parallel surgery processing under uncertainty. *INFORMS Journal on Computing*, 23(2):220–237, 2011. cited By 41.

[21] D.P. Bertsekas and D.A. Castañon. Rollout algorithms for stochastic scheduling problems. *Journal of Heuristics*, 5(1):89–108, 1999. cited By 155.

[22] Papiya Bhattacharjee and Pradip Kumar Ray. Patient flow modelling and performance analysis of healthcare delivery processes in hospitals: A review and reflections. *Computers and Industrial Engineering*, 78:299–312, 2014.

[23] Burak Bilgin, Peter Demeester, Mustafa Misir, Wim Vancroonenburg, and Greet Vanden Berghe. One hyper-heuristic approach to two timetabling problems in health care. *Journal of Heuristics*, 18(3):401–434, June 2012.

[24] Gabriel R. Bitran and Reinaldo Morabito. Open queueing networks: Optimization and performance evaluation models for discrete manufacturing systems. *Production and Operations Management*, 5(2):163–193, 1996.

[25] J.T. Blake and M.W. Carter. Surgical process scheduling: a structured review. *Journal of the Society for Health Systems*, 5(3):17–30, 1997. cited By 59.

[26] Bloomberg. Bloomberg best (and worst). `https://www.bloomberg.com/graphics/best-and-worst/#most-efficient-health-care-2014-countries`. Accessed: 04-06-2018.

[27] Nardo Jonathan Borgman. *Managing urgent care in hospitals*. PhD thesis, University of Twente, Enschede, The Netherlands, 2017.

[28] Richard J. Boucherie and Nico M. van Dijk. *Queueing Networks: A Fundamental Approach*. Springer, 2011.

[29] James R. Broyles, Jeffery K. Cochran, and Douglas C. Montgomery. A statistical markov chain approximation of transient hospital inpatient inventory. *European Journal of Operational Research*, 207(3):1645–1657, 2010.

[30] Eduardo Cabrera, Emilio Luque, Manel Taboada, Francisco Epelde, and Ma Luisa Iglesias. Abms optimization for emergency departments. *Proceedings - Winter Simulation Conference*, page 6465116, 2012.

[31] B. Cardoen, E. Demeulemeester, and J. Beliën. Optimizing a multiple objective surgical case sequencing problem. *International Journal of Production Economics*, 119(2):354–366, 2009. cited By 63.

[32] B. Cardoen, E. Demeulemeester, and J. Beliën. Operating room planning and scheduling: A literature review. *European Journal of Operational Research*, 201(3):921–932, 2010.

[33] S. Ceschia and A. Schaerf. Local search and lower bounds for the patient admission scheduling problem. *Computers & Operations Research*, 38(10):1452–1463, October 2011.

[34] Sara Ceschia and Andrea Schaerf. Modeling and solving the dynamic patient admission scheduling problem under uncertainty. *Artificial Intelligence in Medicine*, 56(3):199–205, 2012.

[35] Sara Ceschia and Andrea Schaerf. Dynamic patient admission scheduling with operating room constraints, flexible horizons, and patient delays. *Journal of Scheduling*, 19(4):377–389, aug 2016.

[36] H.S. Chang, M.C. Fu, J. Hu, and S.I. Marcus. An asymptotically efficient simulation-based algorithm for finite horizon stochastic dynamic programming. *IEEE Transactions on Automatic Control*, 52(1):89–94, 2007. cited By 12.

[37] H.S. Chang, R. Givan, and E.K.P. Chong. Parallel rollout for online solution of partially observable markov decision processes. *Discrete Event Dynamic Systems: Theory and Applications*, 14(3):309–341, 2004. cited By 36.

[38] H.S. Chang, R. Givan, and E.K.P. Chong. *Simulation-Based Algorithms for Markov Decision Processes*. Springer, 2013.

[39] B Cheang, H Li, A Lim, and B Rodrigues. Nurse rostering problems - a bibliographic survey. *European Journal of Operational Research*, 151(3):447–460, 2003.

[40] J. K. Cochran and K. Roche. A queuing-based decision support methodology to estimate hospital inpatient bed demand. *Journal of the Operational Research Society*, 59(11):1471–1482, 2008.

[41] Jeffery K. Cochran and Aseem Bharti. Stochastic bed balancing of an obstetrics hospital. *Health Care Management Science*, 9(1):31–45, 2006.

[42] Jeffery K. Cochran and Kevin T. Roche. A multi-class queuing network analysis methodology for improving hospital emergency department performance. *Computers and Operations Research*, 36(5):1497–1512, 2009.

[43] Shane Combs, Rose Chapman, and A. Bushby. Evaluation of fast track. *Accident and Emergency Nursing*, 15(1):40–47, 2007.

[44] Dorsaf Daldoul, Issam Nouaouri, Hanen Bouchriha, and Hamid Allaoui. Optimization on human and material resources in emergency department. pages 633–638, 2015.

[45] C. de la Maisonneuve and J. Oliveira Martins. Public spending on health and long-term care. 2013.

[46] Mieke Defraeye and Inneke Van Nieuwenhuyse. Staffing and scheduling under nonstationary demand for service: A literature review. *Omega-international Journal of Management Science*, 58:4–25, 2016.

[47] P. Demeester, W. Souffriau, P. De Causmaecker, and G. Vanden Berghe. A hybrid tabu search algorithm for automatically assigning patients to beds. *Artificial Intelligence in Medicine*, 48(1):61–70, 2010.

[48] Xinyang Deng, Yong Deng, Xinyang Deng, Qi Liu, and Qi Liu. Newborns prediction based on a belief markov chain model. *Applied Intelligence*, 43(3):473–486, 2015.

[49] B. Denton, J. Viapiano, and A. Vogl. Optimization of surgery sequencing and scheduling decisions under uncertainty. *Health Care Management Science*, 10(1):13–24, 2007. cited By 170.

[50] B.T. Denton, A.S. Rahman, H. Nelson, and A.C. Bailey. Simulation of a multiple operating room surgical suite. pages 414–424, 2006. cited By 39.

[51] E. Erdem, X. Qu, and J. Shi. Rescheduling of elective patients upon the arrival of emergency patients. *Decision Support Systems*, 54(1):551–563, 2012. cited By 10.

[52] H. Fei, N. Meskens, and C. Chu. A planning and scheduling problem for an operating theatre using an open scheduling strategy. *Computers and Industrial Engineering*, 58(2):221–230, 2010. cited By 92.

[53] Z. Feldman, A. Mandelbaum, W.A. Massey, and W. Whitt. Staffing of time-varying queues to achieve time-stable performance. *Management Science*, 54(2):324–338, 2008.

[54] Thomas A. Feo and Mauricio G.C. Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8(2):67–71, 1989.

[55] Organisation for Economic Co-operation and Development. Oecd health statistics 2017 - frequently requested data, 2017.

[56] Avi Giloni and Sridhar Seshadri. Optimal configurations of general job shops. *Queueing Systems*, 39(2-3):137–155, 2001.

[57] F. Glover. Future paths for integer programming and links to artificial-intelligence. *Computers and Operations Research*, 13(5):533–549, 1986.

[58] J Goldman, H A Knappenberger, and J C Eller. Evaluating bed allocation policy with computer simulation. *Health Services Research*, 3(2):119–29, 119–129, 1968.

[59] F. Gorunescu, P. H. Millard, and S. I. McClean. A queueing model for bed-occupancy management and planning of hospitals. *Journal of the Operational Research Society*, 53(1):19–24, 2002.

[60] Florin Gorunescu, Sally I. McClean, and Peter H. Millard. Using a queueing model to help plan bed allocation in a department of geriatric medicine. *Health Care Management Science*, 5(4):307–312, 2002.

[61] Winfried Grassmann. Transient solutions in markovian queues. *European Journal of Operational Research*, 1(6):396–402, 1977.

[62] LV Green. How many hospital beds? *Inquiry-the Journal of Health Care Organization Provision and Financing*, 39(4):400–412, 2002.

[63] Donald Gross and Carl M. Harris. *Fundamentals of Queueing Theory*. John Wiley & Sons, 1974.

[64] F. Guerriero and R. Guido. Operational research in the management of the operating theatre: A survey. *Health Care Management Science*, 14(1):89–114, 2011.

[65] Randolph W. Hall. *Patient Flow: Reducing Delay in Healthcare Delivery*. Springer US, 2006.

[66] H. Hansagi, B. Carlsson, and B. Brismar. The urgency of care need and patient satisfaction at a hospital emergency department. *Health Care Management Review*, 17(2):71–75, 1992. cited By 59.

[67] R. A. Harris. Hospital bed requirements planning. *European Journal of Operational Research*, 25(1):121–126, 1984.

[68] L. He, Y. Li, and S.H. Chung. Markov chain based modeling and analysis of colonoscopy screening processes. pages 740–745, 2017.

[69] Liselotte Hojgaard. *Hvordan Får Vi Verdens Bedste Sundhedsvæsen?* Informations Forlag, 2017.

[70] Steffen Jacobsen. *Hvis De lige vil sidde helt stille, frue, dr. Jacobsen er ny på afdelingen*. Lindhardt og Ringhof, 2018.

[71] Ulf Kåre Jansbøl. Sundhedsplatformen giver måske en bedre behandling – men den presser og stresser læger og sygeplejersker, 2017.

[72] E.C. Jauch, J.L. Saver, H.P. Adams, A. Bruno, J.J.B. Connors, B.M. Demaerschalk, P. Khatri, P.W. McMullan Jr., A.I. Qureshi, K. Rosenfield, P.A. Scott, D.R. Summers, D.Z. Wang, M. Wintermark, and H. Yonas. Guidelines for the early management of patients with acute ischemic stroke: A guideline for healthcare professionals from the american heart association/american stroke association. *Stroke*, 44(3):870–947, 2013.

[73] O.B. Jennings, A. Mandelbaum, W.A. Massey, and W. Whitt. Server staffing to meet time-varying demand. *Management Science*, 42(10):1383–1394, 1996. cited By 131.

[74] David G. Kendall. Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded markov chain. *The Annals of Mathematical Statistics*, 24:338–354, 1953.

[75] Graham Kendall and Edmund K. Burke. *Search Methodologies*. Springer US, 2005.

[76] Mohammed Khadem, Hamdi A. Bashir, Yasin Al-Lawati, and Fatma Al-Azri. Evaluating the layout of the emergency department of a public hospital using computer simulation modeling: A case study. *I C Indus E*, pages 1709–1713, 2008.

[77] Dirk P. Kroese, Tim Brereton, Thomas Taimre, and Zdravko I. Botev. Why the monte carlo method is so important today. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6(6):386–392, 2014.

[78] Laszlo Lakatos, Laszlo Szeidl, and Miklos Telek. *Introduction to Queueing Systems with Telecommunication Applications*. Springer, 2013.

[79] M. Lamiri, X. Xie, A. Dolgui, and F. Grimaud. A stochastic model for operating room planning with elective and emergency demand for surgery. *European Journal of Operational Research*, 185(3):1026–1037, 2008. cited By 127.

[80] Nadia Landex. The epic healthcare system in denmark. *Ugeskrift for Laeger*, 179(50), 2017.

[81] D C Lane, C Monefeldt, and J V Rosenhead. Looking in the wrong place for healthcare improvements: A system dynamics study of an accident and emergency department. *Journal of the Operational Research Society*, 51(5):518–531, 2000.

[82] Marek Laskowski, Robert D. McLeod, Marcia R. Friesen, Blake W. Podaima, and Attahiru S. Alfa. Models of emergency departments for reducing patient waiting times. *PLoS One*, 4(7):Article No.: e6127, 2009.

[83] Xiaodong Li, Patrick Beullens, Dylan Jones, and Mehrdad Tamiz. Optimal bed allocation in hospitals. 2009.

[84] S. Liao, G. Koole, C. van Delft, and O. Jouini. Staffing a call center with uncertain non-stationary arrival rate and flexibility. *OR Spectrum*, 34(3):691–721, 2012.

[85] Morgan E. Lim, Tim Nye, James M. Bowen, Jerry Hurley, Ron Goeree, and Jean-Eric Tarride. Mathematical modeling: The case of emergency department waiting times. *International Journal of Technology Assessment in Health Care*, 28(2):93–109, 2012.

[86] P. L. Madsen and T. E. Christen. Værdibaseret ledelse i sundhedsvæsenet – en primer. *Ugeskrift for Laeger*, 2018.

[87] J.H. May, W.E. Spangler, D.P. Strum, and L.G. Vargas. The surgical scheduling problem: Current research and future opportunities. *Production and Operations Management*, 20(3):392–405, 2011.

[88] L. Mayhew and D. Smith. Using queuing theory to analyse the government's 4-h completion time target in accident and emergency departments. *Health Care Management Science*, 11(1):11–21, 2008.

[89] Christelijke Mutualiteit. Twaalfde CM-ziekenhuisbarometer (*English: Twelfth CM-hospital barometer*).

[90] J.R. McMillan, M.S. Younger, and L.C. De Wine. Satisfaction with hospital emergency department as a function of patient triage. *Health Care Management Review*, 11(3):21–27, 1986. cited By 77.

[91] D. J. Medeiros, Eric Swenson, and Christopher DeFlitch. Improving patient flow in a hospital emergency department. *2008 Winter Simulation Conference, Vols 1-5*, pages 1526–1531, 2008.

[92] D. Min and Y. Yih. Scheduling elective surgery under uncertainty and downstream capacity constraints. *European Journal of Operational Research*, 206(3):642–652, 2010. cited By 53.

[93] Richard J. Mullins and N. Clay Mann. Population-based research assessing the effectiveness of trauma systems. *Journal of Trauma: Injury, Infection, and Critical Care*, 47(SUPPLEMENT):S59–S66, 1999.

[94] H.P. Newsholme. Hospital bed accommodation. *Public Health - the Journal of the Society of Medical Officers of Health*, 46:73–77, 1932.

[95] OECD. Competition in hospital services, 2012.

[96] National Audit Office of Denmark. Beretning til statsrevisorerne om hospitalernes brug af personaleresurser. Available at http://rigsrevisionen.dk/publikationer/2015/102014/.

[97] Ministry of Health. Status paa sundhedsomraadet. Available at http://www.sum.dk/Aktuelt/Publikationer/Status-paa-sundhedsomraadet-sept-2015.aspx.

[98] World Health Organization. World health report - health systems financing: The path to universal coverage, 2010.

[99] Ronny M. Otero, H. Bryant Nguyen, David T. Huang, David F. Gaieski, Munish Goyal, Kyle J. Gunnerson, Stephen Trzeciak, Robert Sherwin, Christopher V. Holthaus, Tiffany Osborn, and Emanuel P. Rivers. Early goal-directed therapy in severe sepsis and septic shock revisited - concepts, controversies, and contemporary findings. *Chest*, 130(5):1579–1595, 2006.

[100] J. Peck and S. Kim. Improving emergency department patient flow through optimal fast track usage. *Annals of Emergency Medicine*, 52(4):S88–S88, 2008.

[101] Petersen Niels Chr. Pedersen, Kjeld Møller, editor. *Fremtidens Hospital*. Munksgaard, 2014.

[102] J F Pendergast and W B Vogel. A multistage model of hospital bed requirements. *Health Services Research*, 23(3):381–399, 1988.

[103] Luiz Ricardo Pinto, Francisco Carlos Cardoso de Campos, Ignez Helena Oliva Perpetuo, and Yara Cristina Neves Marques Barbosa Ribeiro. Analysis of hospital bed capacity via queuing theory and simulation. pages 1281–1292, 2014.

[104] Martin L. Puterman. *Markov decision processes :*. Wiley, 2005.

[105] Ramandeep S. Randhawa. Optimality gap of asymptotically derived prescriptions in queueing systems o(1)-optimality. *Queueing Systems*, 83(1-2):131–155, 2016.

[106] T.M. Range, D. Kozlowski, and N.C. Petersen. Dynamic job assignment: A column generation approach with an application to surgery allocation. *Discussion Papers on Business and Economics*.

[107] Troels Martin Range, Richard Martin Lusby, and Jesper Larsen. A column generation approach for solving the patient admission scheduling problem. *European Journal of Operational Research*, 235(1):252–264, May 2014.

[108] Andreas Rudkjøbing. Endelig kan vi sige farvel til produktivitetskravet. *Ugeskrift for Laeger*, 179(21):1801, 2017.

[109] M. Samudra, C. Van Riet, E. Demeulemeester, B. Cardoen, N. Vansteenkiste, and F.E. Rademakers. Scheduling operating rooms: achievements, challenges and pitfalls. *Journal of Scheduling*, 19(5):493–525, 2016.

[110] Paul J. Sanchez. Fundamentals of simulation modeling. *Proceedings - Winter Simulation Conference*, pages 4419588, 54–62, 2007.

[111] Robert Schmidt, Sandra Geisler, and Cord Spreckelsen. Decision support for hospital bed management using adaptable individual length of stay estimations and shared resources. *BMC medical informatics and decision making*, 13(1):3, January 2013.

[112] Thomas J. Schriber, Daniel T. Brunner, and Jeffrey S. Smith. Inside discrete-event simulation software. *Proceedings of the 2015 Winter Simulation Conference*, pages 1–15, 2015.

[113] Justus Arne Schwarz, Gregor Selinka, and Raik Stolletz. Performance analysis of time-dependent queueing systems: Survey and classification. *Omega-international Journal of Management Science*, 63:170–189, 2016.

[114] Sridhar Seshradi and Michael Pinedo. Optimal allocation of resources in a job shop environment. *Iie Transactions Industrial Engineering Research and Development*, 31(3):195–206, 1999.

[115] X. Shao, J. Li, and D.A. Wiegmann. A markov chain approach to study flow disruptions on surgery in emergency care. pages 990–995, 2013.

[116] B. Shaw and A. H. Marshall. Modelling the flow of congestive heart failure patients through a hospital system. *Journal of the Operational Research Society*, 58(2):212–218, 2007.

[117] Robin Sibson. *A Brief Description of Natural Neighbor Interpolation*. John Wiley and Sons, 1981.

[118] D Sinreich and Y Marmor. Emergency department operations: The basis for developing a simulation tool. *Iie Transactions*, 37(3):233–245, 2005.

[119] David Sinreich, Ola Jabali, and Nico P. Dellaert. Reducing emergency department waiting times by adjusting work shifts considering patient visits to multiple care providers. *Iie Transactions Industrial Engineering Research and Development*, 44(3):163–180, 2012.

[120] J.M. Smith, F.R.B. Cruz, and T. Van Woensel. Optimal server allocation in general, finite, multi-server queueing networks. *Applied Stochastic Models in Business and Industry*, 26(6):705–736, 2010.

[121] K. Steins, F. Persson, and M. Holmer. Increasing utilization in a hospital operating department using simulation modeling. *Simulation*, 86(8-9):463–480, 2010. cited By 23.

[122] John D. Sterman. System dynamics modeling: Tools for learning in a complex world. *California Management Review*, 43(4):8–25, 2001.

[123] William J. Stewart. *Probability, Markov Chains, Queues, and Simulation - The Mathematical Basis of Performance Modeling*. Princeton University Press, 1 edition, 2009.

[124] Alan B. Storrow, Chuan Zhou, Gary Gaddis, Jin H. Han, Karen Miller, David Klubert, Andy Laidig, and Dominik Aronsky. Decreasing lab turnaround time improves emergency department throughput and decreases emergency medical services diversion: A simulation model. *Academic Emergency Medicine*, 15(11):1130–1135, 2008.

[125] David Y. Sze. Queueing model for telephone operator staffing. *Operations Research*, 32(2):229–249, 1984.

[126] J.-S. Tancrez, B. Roland, J.-P. Cordier, and F. Riane. How stochasticity and emergencies disrupt the surgical schedule. *Studies in Computational Intelligence*, 189:221–239, 2009. cited By 3.

[127] D. Tipper and M.K. Sundareshan. Numerical methods for modeling computer networks under nonstationary conditions. *IEEE Journal on Selected Areas in Communications*, 8(9):1682–1695, 1990.

[128] C. Van Huele and M. Vanhoucke. Analysis of the integration of the physician rostering problem and the surgery scheduling problem. *Journal of medical systems*, 38(6):43, 2014. cited By 4.

[129] J.M. Van Oostrum, M. Van Houdenhoven, J.L. Hurink, E.W. Hans, G. Wullink, and G. Kazemier. A master surgical scheduling approach for cyclic scheduling in operating room departments. *OR Spectrum*, 30(2):355–374, 2008. cited By 83.

[130] P.T. Vanberkel and J.T. Blake. A comprehensive simulation for wait time reduction and capacity planning applied in general surgery. *Health Care Management Science*, 10(4):373–385, 2007. cited By 74.

[131] Wim Vancroonenburg, Patrick De Causmaecker, and Greet Vanden Berghe. A study of decision support models for online patient-to-room assignment planning. *Annals of Operations Research*, 2013. Available online.

[132] Wim Vancroonenburg, Federico Della Croce, Dries Goossens, and Frits C. R. Spieksma. The Red-Blue transportation problem. *European Journal of Operational Research*, 237(3):814–823, 2014.

[133] Richard Varga. p-cyclic matrices: A generalization of the young-frankel successive overrelaxation scheme. *Pacific Journal of Mathematics*, 9(2):617–628, 1959.

[134] Annemie Verhelst. Opnameplanning in ziekenhuizen : 10 jaar later (*English: Admission scheduling in hospitals: 10 years later*), school = Universiteit Gent, year = 2009,. Master's thesis.

[135] Junwen Wang, Xiang Zhong, Jingshan Li, and Patricia Kunz Howard. Modeling and analysis of care delivery services within patient rooms: A system-theoretic approach. *Ieee Transactions on Automation Science and Engineering*, 11(2):6471257, 379–393, 2014.

[136] Lu Wang. An agent-based simulation for workflow in emergency department. pages 19–23, 2009.

[137] X. Wang. Emergency department staffing: A separated continuous linear programming approach. *Mathematical Problems in Engineering*, 2013, 2013.

[138] W. Whitt. Staffing a call center with uncertain arrival rate and absenteeism. *Production and Operations Management*, 15(1):88–102, 2006.

[139] F WILCOXON. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[140] Worldometers. Population. `http://www.worldometers.info/population/`. Accessed: 31-05-2018.

[141] W. Xiang, J. Yin, and G. Lim. A short-term operating room surgery scheduling problem integrating multiple nurses roster constraints. *Artificial Intelligence in Medicine*, 63(2):91–106, 2015. cited By 2.

[142] Xiaolei Xie, Jingshan Li, Colleen H. Swartz, and Yue Dong. Modeling and analysis of hospital inpatient rescue process: A markov chain approach. *Ieee International Conference on Automation Science and Engineering*, pages 6653987, 978–983, 2013.

[143] JY Yeh and WS Lin. Using simulation technique and genetic algorithm to improve the quality care of a hospital emergency department. *Expert Systems With Applications an International Journal*, 32(4):1073–1083, 2007.

[144] K. Yoneda, I. Wada, and K. Haruki. Job shop configuration with queueing networks and simulated annealing. pages 407–410, 1992.

[145] David Young. Iterative methods for solving partial difference equations of elliptic type. *Transactions of the American Mathematical Society*, 76(1):92–111, 1954.

[146] Hojjat Zeraati, Farid Zayeri, Gholamreza Babaee, Navid Khanafshar, and Fatemeh Ramezanzadeh. Required hospital beds estimation: A simulation study. 2005.

[147] Xin Li Zhang, Ting Zhu, Li Luo, Chang Zheng He, Yu Cao, and Ying Kang Shi. Forecasting emergency department patient flow using markov chain. *2013 10th International Conference on Service Systems and Service Management - Proceedings of Icsssm 2013, Int. Conf. Serv. Syst. Serv. Manage.*, pages 6602537, 278–282, 2013.

Appendices

Chapter A

Appendix for Chapter 3

A.1 Equations

Derivative of the Erlang-B formula,

$$\frac{dB}{dM_i} = -\frac{\left(\frac{\lambda_i}{\mu_i}\right)^{M_i+1}\Gamma(M_i+1)\left(\frac{\lambda_i}{\mu_i}\right)^{M_i}\left(G_{2,3}^{3,0}\left(\begin{smallmatrix}1,1\\0,0,M_i+1\end{smallmatrix}\middle|\frac{\lambda_i}{\mu_i}\right)\left(\frac{\lambda_i}{\mu_i}\right)^{M_i}\lambda_i+\Gamma(M_i+1)\left(\Psi(M_i+1)-ln\left(\frac{\lambda_i}{\mu_i}\right)\right)\right)\left(\mu_i\left(\frac{\lambda_i}{\mu_i}\right)^{M_i+1}-\left(\frac{\lambda_i}{\mu_i}\right)^{M_i}\lambda_i\right)\mu_i}{e^{\lambda_i/\mu_i}\ M_i!\left(\left(\frac{\lambda_i}{\mu_i}\right)^{M_i+1}\mu_i\Gamma(M_i+1)+\lambda_i\left(\frac{\lambda_i}{\mu_i}\right)^{M_i}\left(\Gamma\left(M_i+1,\frac{\lambda_i}{\mu_i}\right)-\Gamma(M_i+1)\right)\right)^2} \tag{A.1}$$

where $G_{p,q}^{m,n}\left(\begin{smallmatrix}a_1,\ldots,a_n,a_{n+1},\ldots,a_p\\b_1,\ldots,b_m,b_{m+1},\ldots,b_q\end{smallmatrix}\middle|\, z\right)$ is the Meijer-G function, $\Gamma(x)$ and $\Gamma(s,x)$ the complete and upper incomplete gamma functions, respectively; and $\Psi(x)$ the digamma function.

A.2 Figures

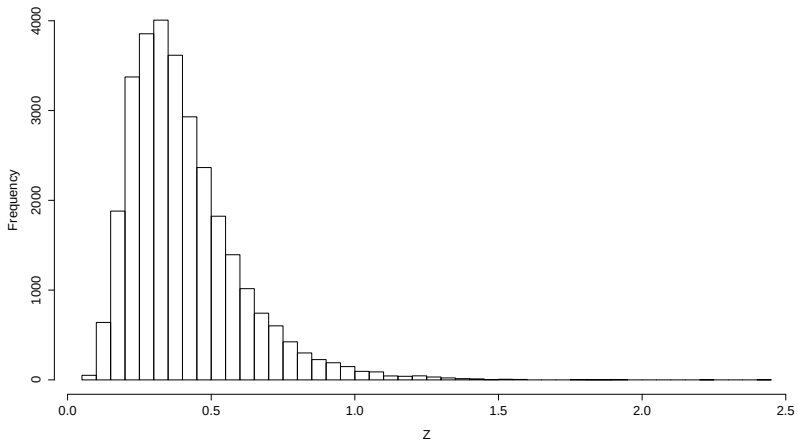


Figure A.1: Simulated distribution of (3.9). Conducted with 30,000 replications.

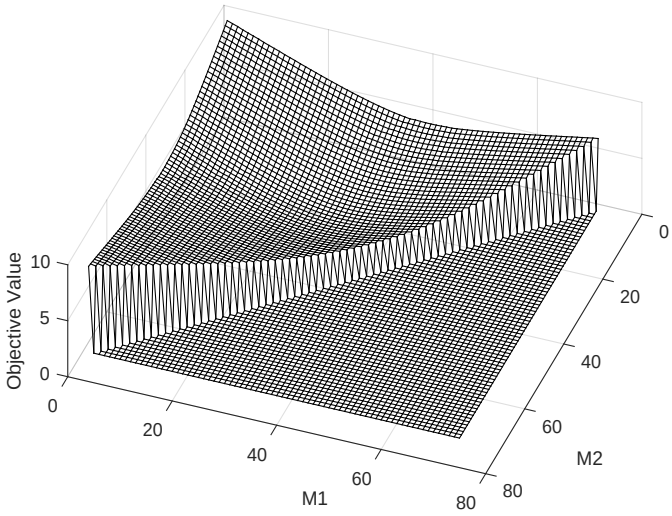


Figure A.2: Complete enumeration of the search space for the current distribution of beds.

Chapter B

Appendix for Chapter 5

B.1 Parameters

Staff Types	Service Times (h)
Triage Nurse	$1/6$
Basic Physicians	$1/3$
Specialized Medical Physicians	$3/4$
Organ Surgeons	$3/4$
Orthopedic Surgeons	$3/4$

Table B.1: Assumed average service times for each of the five staff types.

From \ To	1	2	3	4	5	Discharge
1		1.00				
2		0.10	0.53	0.25	0.11	0.01
3			0.50			0.50
4				0.50		0.50
5					0.50	0.50

Table B.2: Routing probabilities for the queueing network presented in Figure 5.1.

Staff Type	Waiting Time Target (h)
Triage Nurse	$1/6$
Basic Physician	1
Specialized Medical Physician	3
Organ Surgeons	3
Orthopedic Surgeons	3

Table B.3: Waiting time targets, ν_c, used to evaluate the performance of the ED.

Variation	l	a	p_f	y
add-remove	15	5	0.75	40
move-remove	15	5	0.75	10,000

Table B.4: Parameters used in the Recursive Bound Adaptation tests.

B.2 Algorithms

Algorithm 14 The tabu search heuristic.

$x_{cj} \leftarrow INITIALIZE(), L \leftarrow \emptyset$ ▷ Initialize solution x_{cj} and tabu list L
$x^*_{cj} \leftarrow x_{cj}, f^* \leftarrow EVALUATE(x^*_{cj})$
while $elapsedtime < maxtime$ **do**
    $N \leftarrow CREATE(x_{cj}, p_f, a)$ ▷ Create neighborhood of size a, using solution x_{cj} and fraction p_f
    $j \leftarrow 1, b \leftarrow N[j]$
    for $i = 2$ to $|N|$ **do** ▷ Find the best solution in the neighborhood
        $f \leftarrow EVALUATE(N[i])$
        if $f < b$ and ($N[i] \notin L$ or $f < f^*$) **then**
            $b \leftarrow f, j \leftarrow i$
        end if
    end for
    $x_{cj} \leftarrow N[j], L \leftarrow UPDATE(N[j], l)$ ▷ Move to the best permissible solution and update the tabu list
    if $f < f^*$ **then**
        $f^* \leftarrow f, x^*_{cj} \leftarrow x_{cj}$ ▷ Save the best known solution
    end if
end while
    return x^*_{cj}